Engineering Handbuch

für die Bausteinbibliothek

Sm@rtLib

V4.1

Stand 14.02.2011

Vorwort

Sm@rtLib ist Produkt der Firma AGU und wird von der Firma Siemens als Premium AddOn unterstützt (Bereich HVAC). Diese Bausteinbibliothek kann für PCS7, WinCC, WinCCflexible, S7-300 und S7-400 Systeme eingesetzt werden. Sm@rtLib realisiert den Gedanken der vertikalen Integration aller aufgeführten Komponenten.

Die Durchgängigkeit im Bereich der Melde- und Quittierverarbeitung und die effektive Projektierung sind Garant für eine schnelle Einarbeitung in dieses Produkt.

Die Funktionen der Bibliothek erlauben den Einsatz in unterschiedlichen Branchen. Automatisierungstechnik aus dem Baukasten heißt die Philosophie und aus diesem Grunde sind die Funktionen so gegliedert, dass sich der Anwender seine Master-Loops und Standardfunktionen aus dem vorhandenen Vorrat zusammen bauen kann.

Unsere Erfahrung in der Erstellung von Standards beruht auf 15 Jahren erfolgreicher Zusammenarbeit mit großen Firmen in der Pharmaindustrie, Chemie, Gebäude- und Fertigungstechnik.

Ziel dieses Dokuments

In diesem Dokument werden detailliert die Projektierungsschritte eines Step7 oder PCS7 Projektes mit SmartLib beschrieben. Es enthält allgemeine Informationen zur Projektierung und Hintergrundinformationen zu SmartLib. Eine detaillierte Beschreibung der einzelnen Bausteine erhalten Sie wenn Sie im Simatic Manager oder in CFC mit „F1“ die Kontexthilfe aufrufen.
Grundsätzliche Erläuterungen zu STEP7 , PCS7, WinCC oder WinCCflexible entnehmen Sie bitte den entsprechenden Produkthandbüchern von SIEMENS.

Bausteinübersicht

Bezeichnung	FB Nr.	Funktion	Function
A_INPUT_41	FB571	Messwert mit 4 Grenzwerten	Analoginput
A_OUTPUT_41	FB572	Stellglied analog (z.B. Regelventil)	Analogoutput
ACTUATOR_41	FB553	Stellglied binär (Regelklappe)	Actuator
AEM_VALVE_41	FB576	Zuluft-, Abluft-, Mischluftklappe	Additional- Exhaust- Mixed Air Valve
C_CTRL_41	FB573	Kontinuierlicher PID-Regler	PID-Controler
CLOCK_TIMER_41	FB579	Schaltuhr / Wochenprogramm	Clock Timer
CONVERTER_41	FB560	Frequenzumrichter	Frequency Converter
DATETIME_41	FB580	Systemzeit und Datum	Date and Time Module
DIFF_INT_41	FB575	Summierer und Differenzierer	Differentiator Integrator
DOS_41	FB574	Dosierer	Doser Module
FW_CONTROL_41	FB582	Schrittkettenbetriebsarten	Step sequence modes
HT_CURVE_41	FB584	Heizkennlinie	Heating Curve
HX_CURVE_41	FB585	HX-Kennlinie (Feuchte)	Dewpoint calculation
INTERPOL_41	FB570	Polygonzug	Interpolate module
LIMIT_41	FB569	Grenzwertschalter analog	Limit Switch
M_CONT400_41	FB568	Binärmeldung (nur S7-400)	Threshold Switch (S7-400)
M_CONTACT_41	FB567	Binärmeldung (für S7-300)	Threshold Switch
MOTOR_1N_41	FB552	Motor eine Drehzahl	Motor Single rev
MOTOR_2N_41	FB554	Motor zwei Drehzahlen	Motor Double Speed
MOTOR_LR_41	FB555	Motor Links-Rechtslauf	Motor Double Rev
RTM_41	FB583	Betriebsstundenzähler	Run-time Meter
S_CTRL_41	FB557	Schrittregler	Step-Controler
SCHEDULE_41	FB578	Zeitsteuerung / Wochenprogramm	Scheduler
SPLIT_R_41	FB577	Split-Range Funktion	Split Range Module
SYSTEM	FB551	Systemfunktionen für SmartLib	Common system functions
VALVE_41	FB556	Ventil Auf/Zu	Valve

Inhaltsverzeichnis:

1 Versionshistorie 7

2 Installationshinweise 10

2.1 Software-Voraussetzungen 10

2.2 Installationshinweise von Sm@rtLib 10

3 Allgemeine Hinweise 15

3.1 Auswahl der Programmiersprache 15

3.2 Auswahl der CPU 15

4 Einrichten des Projekts 18

4.1 Vorraussetzungen 18

4.2 Einfügen der Bausteine und Pläne in das S7-Projekt 18

4.2.1 Verbindungsname 18

4.2.2 Sm@rtLib Bausteine undTextlibrary einfügen 18

4.2.3 Systemplan und Musterpläne 19

4.3 WinCC OS konfigurieren 20

4.3.1 Konfigurator 20

4.3.2 Alarm-Logging anpassen 21

4.3.3 Sammelanzeigen anpassen 22

4.3.4 Bedienberechtigungen festlegen 23

5 Projektieren eine Messstelle 24

5.1 Anlegen und konfigurieren der Messstelle in CFC 24

5.2 Anlegen und konfigurieren der Messstelle in WinCC 25

6 Systembausteine: 26

6.1 SYSTEM 26

6.2 MSG_KERNEL 26

6.3 OM_EVENT_MSG 27

6.4 AI_KERNEL 27

6.5 AO_KERNEL 27

6.6 CONDITIONS 27

6.7 Besonderheit Systembausteine in einem AWL-Programm 28

6.8 Systembausteine in einem CFC-Programm 29

7 Meldesystem 30

7.1 Meldefähigkeit mit dem Alarm Baustein der CPU 30

7.2 Meldesystem – S7-400 Steuerung 31

7.3 Meldesystem – S7-300 Steuerung 32

7.3.1 AWL Programme 32

7.4 Meldesystem einstellen 33

8 Quittierungskonzept 36

8.1 Quittierung Status 36

9 WinCC Grundlagen – ab SmartLib V4 37
9.1 Festlegung der Symbole ab SmartLib V4.1.0 37
9.2 Verwendung von ActiveX-Controls 37
9.3 Anwahlbutton für den Faceplate Aufruf für WinCC 39
9.4 Fließbildsymbol für WinCC 43
9.5 Ausgabefeld für WinCC 45
10 WinCC Grundlagen – ab SmartLib V4.1.0 47
10.1 Aufruftaste für das Faceplate 47
11 WinCC Bildsymbole als UserDefinedObjects (UDO) 49
11.1 Bildsymbole verwenden 49
11.2 Grundlagen zum UDO 50
11.2.1 UDO: Beispiel - Motorbaustein 50
11.2.2 UDO: Motorbaustein – Aufbau des UDO 51
11.2.3 UDO: Motorbaustein – interne Funktionen 52
12 Bedienmeldungen für WinCC 64
12.1 Bedienmeldungen aus dem S7-Baustein 64
12.1.1 OS-ID einrichten 64
12.2 Bedienmeldungen über WinCC (ab V4.1.1.3) 64
12.2.1 Bedienung einer Funktion ohne Wertänderung: 64
12.2.2 Bedienung mit Wertänderung 65
12.2.3 Texte in Bedienmeldungen 65
13 Sprachumschaltung projektweit 66
13.1 Neue Sprache für WinCC Faceplates ergänzen 66
Ordner: ...\Siemens\WinCC\Sm@rtLib\Common\Language 66
14 Automatisches Erzeugen von Bildsymbolen im PCS7 68
15 Darstellungen und Bedienung 69
15.1 Störanzeige in WinCC 69
15.2 Allgemeine Anzeigen im Faceplate 71
15.2.1 Grundaufbau 72
15.2.2 Bedienblock und Betriebsartenteil 73
15.2.3 Karteireiter 74
15.2.4 Werteingabe 77
15.3 Analogeingabe 78
15.3.1 Anzeige Istwertgraph 79
15.4 Analogausgabe 80
15.5 Motor 81
15.6 Ventil 82
15.7 Regler 83
15.7.1 Bedienblock 83
15.7.2 Statusfenster 84
15.7.3 Servicefenster 85
15.8 Zähler 86
16 Erstellung von Grundfunktionen 86

17 WinCC flexible 87

17.1 SmartLib WinCC flexible Option zu einem Projekt hinzufügen 88

17.2 Verbindung projektieren 88

Abschließend sollte das WinCC flexible – Projekt gespeichert und neu generiert werden. 89

17.3 Bilder 90
17.3.1 Prozessbild 90
17.3.2 Kreisbild 90
17.3.3 Systembild 91

17.4 Meldesystem 91

17.5 Start der Projektierung und Voraussetzungen 92
17.5.1 Fliessbild erstellen 93
17.5.2 Fliessbildvariablen anlegen 93

17.6 Visualisierung von Aktoren 94
17.6.1 Variablenverbindung 94
17.6.2 Faceplate Anwahl dynamisieren 96

17.7 Visualisierung von Werten 97
17.7.1 Variablenverbindung 97
17.7.2 Faceplate Anwahl dynamisieren 100

17.8 Besonderheiten bei der Projektierung mit CFC 101
17.8.1 GetIDB/ListIDB 101
17.8.2 Fliessbildvariablen anlegen 101
17.8.3 Faceplatevariable im Ordner FP_COMMON anpassen 102
17.8.4 Faceplate Anwahlskript anpassen 102
17.8.5 Faceplate Anwahl dynamisieren 103

18 Anhang 104

18.1 Programmerstellung mit CFC 104

18.2 Programmerstellung mit AWL/FUP/KOP 106

18.3 Bedienberechtigungsstufen für Faceplateanwahl (z.B. UDO) 107

18.4 Belegung der Zustandsworte Q_STATE 110

18.5 Bediennummern 112

18.6 Festlegungen zur Visualisierung 114

1 Versionshistorie

V4.1.0.0

Neben einigen Fehlerbereinigungen und Performanceverbesserungen wurden folgende Punkte bearbeitet

Baustein Regler C_CTLR Eingang für Umschaltung Stellen/Regeln extern eingefügt
Überarbeitung der Kommentare für die S7-Baustein Ein/Ausgänge
Bibliothek für die S7-400 unter Nutzung der vollen Meldfunktionen einer S7-400 CPU. Die Bibliotheken für S7-300 und 400 unterscheiden sich nur durch die Meldverarbeitung der S7-Bausteine.
Neue Bausteine :
ACTUATOR: Steuerung eines elektrisch angetriebenen Stellantriebes (z.B. AUMA-Klappe)
MOTOR_2N: Motor mit zwei Drehzahlen
MOTOR_LR: Motor mit Drehrichtung Links/Rechts
CLOCK_TIMER: Zeitschaltuhr mit Wochenzyklus
S_CTRL: Schrittregler
Für die Symbole im WinCC-Bild gibt es alternativ zu den Active-X Symbolen die Möglichkeit Anwender Objekte (UDO) zu verwenden. Entsprechende Symbole werden mitgeliefert. Ändern der Symbole ist möglich. Diese Symbole können auch in AWL eingesetzt werden.
Erweiterung des Bedienprotokolls für S7-400: Bei der S7-400 werden die Bedienungen an den Bausteinen als Bedienmeldung an WinCC gesendet. Dabei wird, neben der eigentlichen Bedienung, dem Altwert und dem Neuwert auch eine OS-Kennung gesendet. Dadurch kann die Bedienung eindeutig einem Bedienterminal zugeordnet werden.
WinCCFlexible Faceplates und Symbole verfügbar
Der Projektkonfigurator kopiert eine angepasste Sammelanzeige in das WinCC-Systemverzeichnis (optional)
S_CTRL Die Eingabe des Parameters Q_DEADB_W erfolgte bisher in %. Dies wurde umgestellt Q_DEADB_W muss jetzt als Absolutwert eingegeben werden
Übersetzung der Runtimetexte (Störtexte, Faceplatetexte) in Englisch.
Das Konfigurationswerkzeug für den GraCs-Designer unterstützt jetzt auch den FW-CONTROL Baustein

Die neue Version ist NICHT Anschlusskompatibel zu den vorherigen S7-Bausteinen V4.0. Bei einem Update der Bausteine in einem bestehenden Programm muss ein Gesamtladen der CPU durchgeführt werden!

V4.1.0.1

Fahrweisenrahmen unterstützt auch selbstbeendende Grundfunktionen

V4.1.0.2

Meldungen bleiben bei der S7-300 Version nicht mehr hängen.

V4.1.0.3

Istwertverschaltung in FW_CTRL UDO eingetragen
Parametrierung der AWL-Texte für FW_CTRL in UDO Konfigurator eingetragen
S_CTRL Parameter für Regler werden mit Nachkommastellen angezeigt.

V4.1.0.4

Fehler beim Übetragen der UDO Scripte (aguSLibUDOState.bmo) behoben.

V4.1.0.5

Installationsproblem bei fehlendem Microsoft Addin Designer behoben.

V4.1.1.0

S_CTRL Fehler bei Grenzwertaktivierung (vertauscht) behoben
Attribute für Prozessobjektsicht aktiviert
Bereichsspezifische Bedienfreigabe für die Faceplates

V4.1.1.1

Rücklesen von Grenzwerten und Parametern in CFC möglich (C_CTRL, S_CTRL, A_INPUT)
FAQ vom 26.6.2008 im Forum dazu beachten!!!"

V4.1.1.2

Split-Range Baustein geändert (Bisher: nur \ / neu: / /)

V4.1.1.3

Bedienmeldungen aus Faceplate eingefügt. Die Bedienmeldungen werden jetzt vom Baustein UND vom Faceplate direkt in das Bedienprotokoll eingetragen. Die Bedienmeldung vom Faceplate enthält den Bedienernamen.
Dynamik Verriegelungstexte bei S_CTRL, MOTOR_LR und MOTOR_2N korrigiert
Neue Baustein M_CONT400 als Alternative zu M_CONTACT speziell für S7-400

V4.1.1.4

Bausteinparameterbezeichungen auch in Englisch verfügbar (Onlinehilfe F1)
Zusätzliche WinCC Bedienmeldungen können abgeschaltet werden Initialisierungsdatei eingefügt
Unterstützung SIMATIC Batchreport. (Nur Zusammen mit Batch Zusatzbausteinen auf Anfrage)
SmartLib Bausteine können jetz auch in PCS7-SFC als Bausteintypen verschaltet werden
Bedienmeldungen der Toggelfunktionen (PLT, WINT/EXT, STELLEN/REGELN) sind jetzt detailierter.
Bedienberechtigungen für ACTUATOR Stellgrösse korrigiert
Eigenschaften der SmartLib Symbole auf Englisch übersetzt
Dokumentation zu Fahrweisen wurde von Engineering Handbuch in die FW_CONTROL Hilfe verschoben
Der ACTUATOR läuft nach STOP oder VERR (IF_ILOC_STOP_THEN_CLS = 0) nicht mehr automatisch an
Die Verriegelungsbedigungen im Faceplate des ACTUATOR sind jetzt dynamisch (Farbumschlag)
Schnittstellenbezeichung C_CTRL und S_CTRL von LH*x*_OP_EN auf HL*x*_OP_EN geändert
Online Hilfe für HX-Curve Baustein eingefügt
Diverse Fehlerbereinigungen

V4.1.1.5

Fehler beim Öffnen des Projektes im WinCC Explorer ab V7.0 SP1 (Progarmmabsturz) behoben.

V4.1.1.6

Serverpräfix bei Client/Server-Betrieb kann jetzt mehr als 20 Zeichen haben (max. 128)
Beim Update von V4.1.1.5 oder kleiner: Faceplateanwahlscripte in allen Bildern übersetzen !! >WinCCExplorer>GraphicsDesigner>Konvertieren der Bilder
Die UDO-Konfigurationsbox verliert nicht mehr die Werte wenn die AWL-Unterbox geöffnet wird

V4.1.1.7

Division durch 0 bei HT_CURVE verhindert
Fehlermeldung bei Faceplateaufruf behoben
(DM_Interface detetction of Blockversion failed <.#blocktype> not found

V4.1.1.8

- Erweiterung der AGU_OS_Template.DLL für den Aufruf der Kormbacher Bauteine C_CTRL_Ex und S_CTRL_Ex

V4.1.1.9

-Erweiterung der AGU_OS_TemplateDLL für den Aufruf des Pumpenmanagers (Optionsbaustein, nicht im Setup enthalten)
-Korrekturen im Engineering Handbuch
-MOTOR_1N externe Quittierung quittiert jetzt auch die Bimetallstörung

V4.1.1.10

-Korrektur der Deklaration der Funktion für das Bedienprotokollin der AGU_OS_Template.DLL
-Neues UDO-Symbolbild: @PCS7Typicals_AGU_SmartLibUDO_FB_Symbole.pdl für die automatische Bildsymbolerzeugung unter PCS7.

2 Installationshinweise

2.1 Software-Voraussetzungen

Um Sm@rtLib betreiben zu können, benötigen Sie auf Ihrem Engineering- bzw. Runtime-System noch folgende Software:

Für das AS/OS-Engineering System:

STEP 7 V5.3 oder höher (erforderlich)
WinCC V6 SP3 oder höher (erforderlich)
CFC V6.1
AS-OS Engineering V6.1

Für das Runtime-System:

WinCC V6 SP3 oder höher (erforderlich)

2.2 Installationshinweise von Sm@rtLib

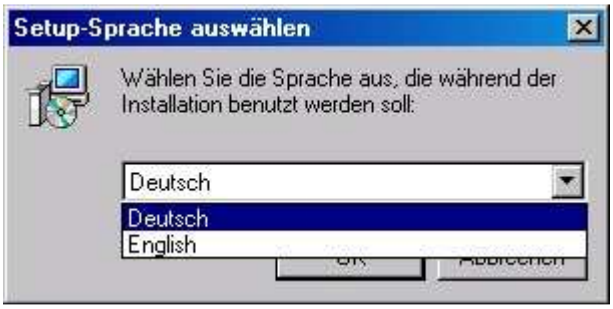
Die Installationshinweise enthalten wichtige Informationen, die Sie für die Installation der Software Sm@rtLib benötigen. Lesen Sie diese Hinweise bitte **vor** der Installation.

Im Folgenden finden Sie die Installationsanleitung für Sm@rtLib.

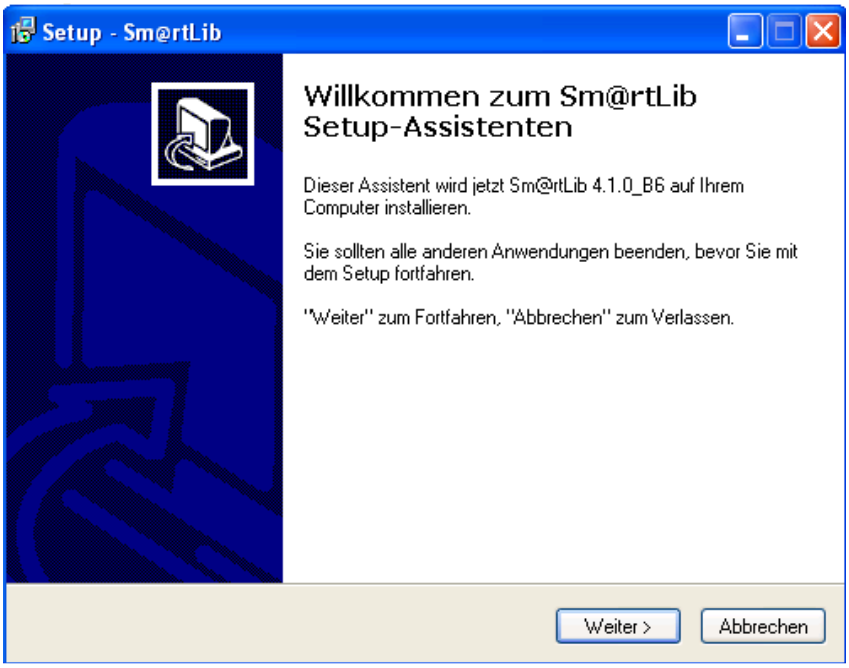
Die Installation umfasst die Softwarebausteine für den Step7 Manager und die Faceplates und Symbole für WinCC (ab V6). Inhalt der Installation sind die S7-Bibliothek, die Visualisierungskomponenten, die Konfigurationstools und die Dokumentation.

Starten Sie das Installationsprogramm Setup.exe.

Wählen Sie die Sprache



Sie werden vom Installationsprogramm begrüßt. Bestätigen Sie das Fenster mit "Weiter".

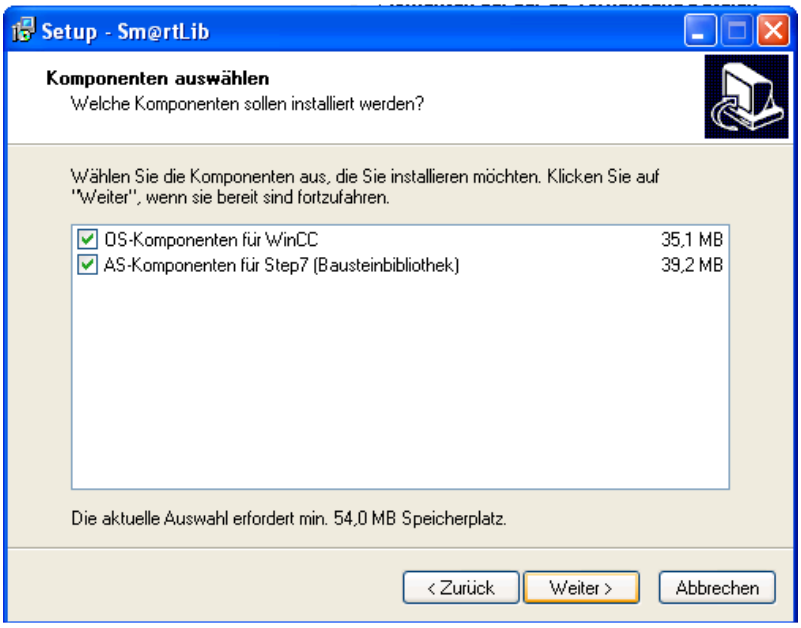

Setup - Sm@rtLib
Willkommen zum Sm@rtLib Setup-Assistenten
Dieser Assistent wird jetzt Sm@rtLib 4.1.0_B6 auf Ihrem Computer installieren.
Sie sollten alle anderen Anwendungen beenden, bevor Sie mit dem Setup fortfahren.
"Weiter" zum Fortfahren, "Abbrechen" zum Verlassen.
Weiter >
Abbrechen

Wählen Sie die Optionen der Software die installiert werden sollen. Für eine Engineering-Station sollten alle Option ausgewählt werden. Für eine Operator-Station sollte die Option OS-Komponenten gewählt werden.

OS-Komponenten: alle erforderlichen Dateien für die Visualisierung (Runtime und Configuration)
Diese Option reicht für eine WinCC-OS aus.

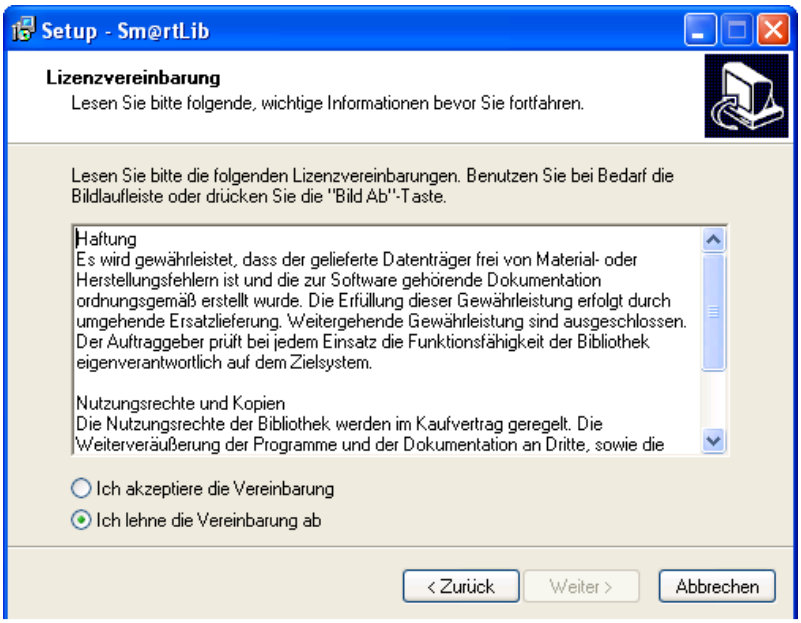
AS-Komponenten: alle erforderlichen Dateien für die Projektierung des S7-Programms
Für die Projektierung ohne Visualisierung reicht diese Option aus.

Examples: Beispielprojekt für die ersten Schritte

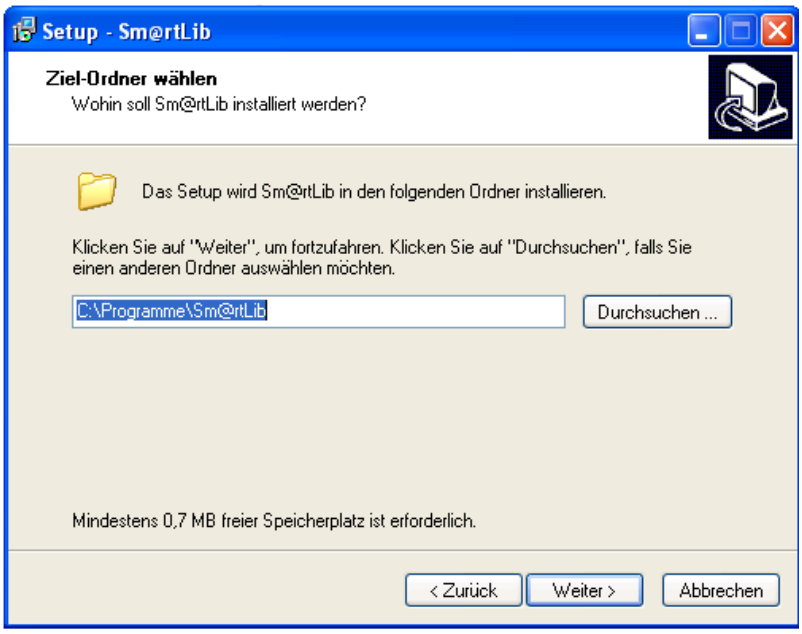

Setup - Sm@rtLib
Komponenten auswählen
Welche Komponenten sollen installiert werden?
Wählen Sie die Komponenten aus, die Sie installieren möchten. Klicken Sie auf "Weiter", wenn sie bereit sind fortzufahren.
OS-Komponenten für WinCC 35,1 MB
AS-Komponenten für Step7 (Bausteinbibliothek) 39,2 MB
Die aktuelle Auswahl erfordert min. 54,0 MB Speicherplatz.
< Zurück
Weiter >
Abbrechen

Bitte beachten Sie die Nutzungsrechte für die Software und bestätigen Sie Ihr Einverständnis mit diesen Bedingungen.

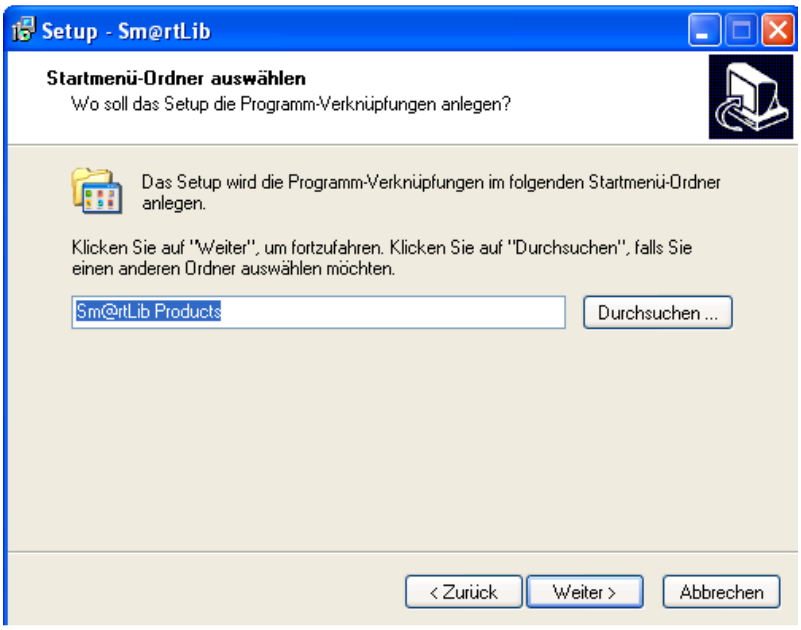
Starten Sie die Installation. Taste WEITER


Setup - Sm@rtLib
Lizenzvereinbarung
Lesen Sie bitte folgende, wichtige Informationen bevor Sie fortfahren.
Lesen Sie bitte die folgenden Lizenzvereinbarungen. Benutzen Sie bei Bedarf die Bildlaufleiste oder drücken Sie die "Bild Ab"-Taste.
Haftung
Es wird gewährleistet, dass der gelieferte Datenträger frei von Material- oder Herstellungsfehlern ist und die zur Software gehörende Dokumentation ordnungsgemäß erstellt wurde. Die Erfüllung dieser Gewährleistung erfolgt durch umgehende Ersatzlieferung. Weitergehende Gewährleistung sind ausgeschlossen. Der Auftraggeber prüft bei jedem Einsatz die Funktionsfähigkeit der Bibliothek eigenverantwortlich auf dem Zielsystem.
Nutzungsrechte und Kopien
Die Nutzungsrechte der Bibliothek werden im Kaufvertrag geregelt. Die Weiterveräußerung der Programme und der Dokumentation an Dritte, sowie die
Ich akzeptiere die Vereinbarung
Ich lehne die Vereinbarung ab
< Zurück
Weiter >
Abbrechen

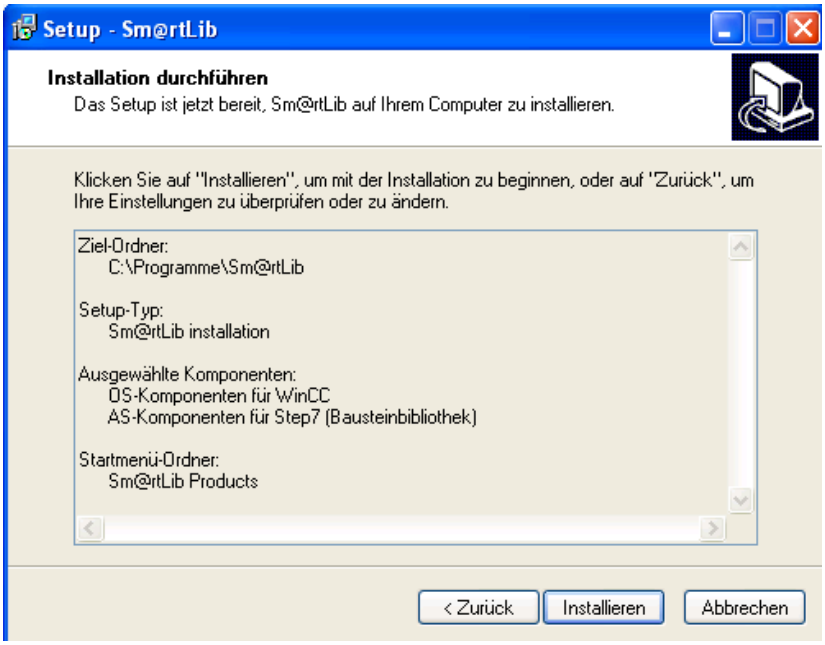
Wählen Sie den Ordner der Ablage aus


Setup - Sm@rtLib
Ziel-Ordner wählen
Wohin soll Sm@rtLib installiert werden?
Das Setup wird Sm@rtLib in den folgenden Ordner installieren.
Klicken Sie auf "Weiter", um fortzufahren. Klicken Sie auf "Durchsuchen", falls Sie einen anderen Ordner auswählen möchten.
C:\Programme\Sm@rtLib
Durchsuchen ...
Mindestens 0,7 MB freier Speicherplatz ist erforderlich.
< Zurück
Weiter >
Abbrechen

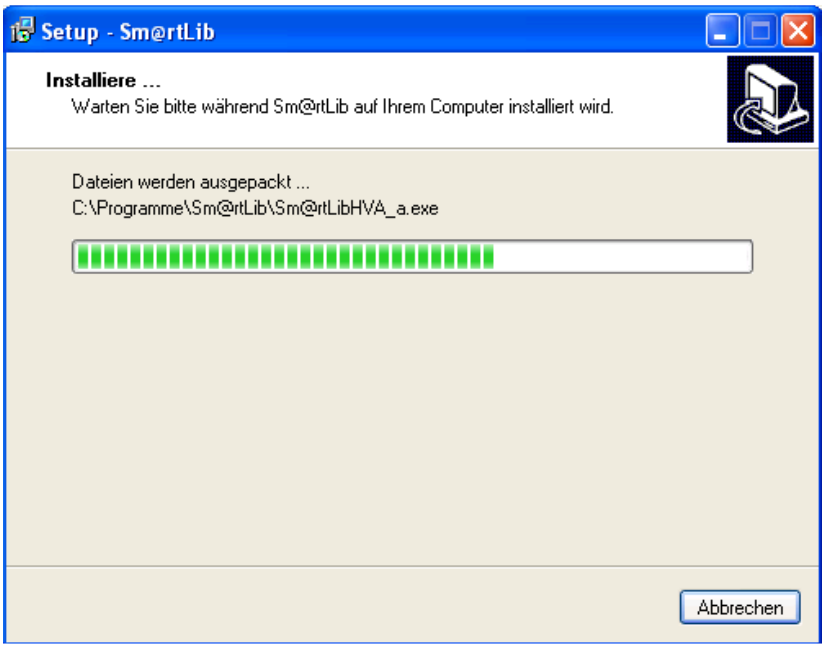
Wählen Sie den Startmenüordner (Programmgruppe)

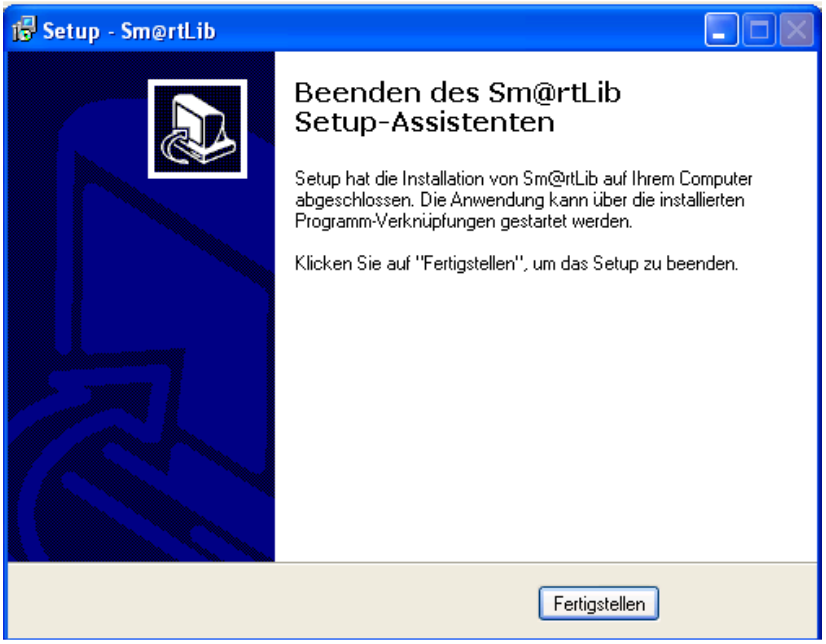

Setup - Sm@rtLib
Startmenü-Ordner auswählen
Wo soll das Setup die Programm-Verknüpfungen anlegen?
Das Setup wird die Programm-Verknüpfungen im folgenden Startmenü-Ordner anlegen.
Klicken Sie auf "Weiter", um fortzufahren. Klicken Sie auf "Durchsuchen", falls Sie einen anderen Ordner auswählen möchten.
Sm@rtLib Products
Durchsuchen ...
< Zurück
Weiter >
Abbrechen

Installationseinstellungen kontrollieren



Start der Installation




Setup - Sm@rtLib
Beenden des Sm@rtLib Setup-Assistenten
Setup hat die Installation von Sm@rtLib auf Ihrem Computer abgeschlossen. Die Anwendung kann über die installierten Programm-Verknüpfungen gestartet werden.
Klicken Sie auf "Fertigstellen", um das Setup zu beenden.
Fertigstellen

Deinstallation von Sm@rtLib

Die Deinstallation von Software-Produkten muss MS Windows-konform erfolgen.

Dazu entfernen Sie über die MS Windows-Applikation "Software" (in der MS Windows Task-Leiste über **Start > Einstellungen > Systemsteuerung > Software**) Ihr Software-Paket (z.B. "Sm@rtLib"). Nur über diesen Weg ist unter MS Windows eine Deinstallation möglich.

Im Rahmen der Deinstallation werden alle Einträge im System gelöscht.

3 Allgemeine Hinweise

3.1 Auswahl der Programmiersprache

Die Nutzung der CFC Programmierumgebung (Optionspaket) **wird empfohlen**, ist aber **nicht** zwingend erforderlich.

Der Anwender kann die Sm@rtLib Funktionsbausteine im Step7 Manager auch in AWL, FUP oder KOP Programmen integrieren. Um den automatischen Variablen-Transfer nach WinCC zu nutzen, sind die Hinweise zur Projektierung von AWL,FUP,KOP Programme zu beachten.

3.2 Auswahl der CPU

Bitte berücksichtigen Sie bei der Auswahl der CPU die Anzahl der meldefähigen Bausteine. Von dieser Anzahl müssen Sie 2 Bausteine abziehen (Sammelanzeige und Bedienmeldung). Im Falle der CPU315 können Sie also 38 Meldefunktionen aufbauen, die zu einer Zeit gleichzeitig melden können. Natürlich können Sie mehr als diese Anzahl an Bausteine einbauen (Speichergrenze).
Die Anzahl der gleichzeitig aktiven AlarmS/SQ Bausteine ist in den technischen Daten der entsprechenden CPU unter dem Punkt *gleichzeitig aktive Alarm-S-Bausteine, max* aufgeführt.

S7-300

CPU 315	40 - 300 Meldefähige Bausteine *[1]
CPU 317	60 – 300 Meldefähige Bausteine *[1]
CPU 319	60 – 300 Meldefähige Bausteine *[1]

*[1] je nach Ausgabestand der CPU

Melderessourcen bei S7-400

Für den Einsatz der Bausteine in einer S7-400 können die SmartLib-Bausteine mit erweiterten Meldebausteinen eingesetzt werden. Da die S7-400 Bausteine den Alarm_8-Baustein verwenden ist bei Einsatz der S7-400 Bausteine die „Anzahl der Kommunikationsaufträge für Alarm_8“ relevant (technische Daten S7-400 CPU).

z.B. CPU 417 maximal 10000 Kommunikationsaufträge für Alarm_8 (einstellbar).
Die Anzahl ist in der Hardwarekonfiguration einstellbar. Beim dem Defaultwert von 1200 Kommunikationsaufträgen können 1200 Meldefähige Bausteine eingebaut werden.

Beachten Sie bitte die Hinweise im Anschnitt 7.1 “Meldefähigkeit mit dem Alarm Baustein der CPU“

Ressourcenbedarf

Die folgenden Tabellen führen die Ressourcenbelegung des einzelnen Sm@rtLib Komponenten auf. Dies kann für die Abschätzung von CPU–Arbeitsspeicher, Melderessourcen und WinCC-Variablenlizenz verwendet werden. Eine Liste als Excel-Datei zur Berechnung finden Sie auf den Webseiten von AGU.
siehe aktuelle Version im Internet www.agu.de

Systembausteine Sm@rtLib

Funktionsbausteine und Funktionen werden von anderen Bausteinen aufgerufen und müssen in jedem AS vorhanden sein.

Nr.	Bausteintyp	Prozessvar.	Arbeitsspeicher FB	Arbeitsspeicher Instanz	Anzahl Meldebst	Option
FB551	SYSTEM	-	740	104	-	Nein
FC560	CONDITIONS	-	1146	-	-	Nein
FC561	TASKNO	-	122	-	-	Nein
FC574	MSGDIAG	-	1104	-	-	Nein
FB562	MSG_KERNEL	-	1714	-	-	Nein
FB563	OM_EVENT_MSG	-	2090	62	2	Nein
DB3	SYSTEMDB	-	38	-	-	Nein
DB4	DIAG_MSG_BUFFER	-	146	-	-	Nein
DB5	EVENT_MSG_BUFFER	-	642	-	-	Nein
DB6	OM_MSG_BUFFER	-	102	-	-	Nein
	**Gesamt Systembausteine*	*0*	***8010***		***2***	

Funktionsbausteine Sm@rtLib

Die Bausteine müssen nur im Bausteinordner vorhanden sein, wenn sie im Programm verwendet werden.

Nr	Bausteintyp	Prozessvar.	Arbeitsspeicher FB	Arbeitsspeicher Instanz S7300	Arbeitsspeicher Instanz S7-400	Anzahl Mel-debst	Option
FB 571	A_INPUT	14	2364	214	342	1	Ja
FB 572	A_OUTPUT	15	3100	234	362	1	Ja
FB 553	ACTUATOR	20	4018	414	542	1	Ja
FB 576	AEM_VALVE	10	526	140	268	1	Ja
FB 573	C_CTRL + CONT_C	36	7566	466	594	1	Ja
FB 557	S_CTRL + CONT_S	32	8212	416	544	1	Ja
FB 560	CONVERTER	26	5294	352	480	1	Ja
FB 580	DATETIME	0	188	60	188	1	Ja*
FB 575	DIFF_INT	10	1154	172	300	1	Ja
FB 584	HT_CURVE	0	272	68	196	1	Ja
FB 570	INTERPOL	0	646	222	222	0	Ja
FB 569	LIMIT	0	376	114	242	1	Ja
FB 567	M_CONTACT	1	886	104	232	1	Ja
FB 568	M_CONT400	1	792	-	278	1	Ja
FB 552	MOTOR_1N	9	2306	218	346	1	Ja
FB 554	MOTOR_2N	9	3858	274	402	1	Ja
FB 555	MOTOR_LR	9	3728	248	376	1	Ja
FB 583	RTM	7	510	130	130	0	Ja
FB 578	SCHEDULE	5	4306	202	202	0	Ja
FB 577	SPLIT_R	16	394	96	96	0	Ja
FB 556	VALVE	9	2070	192	320	1	Ja
FB 574	DOS	25	4012	258	386	1	Ja
FB 579	CLOCK_TIMER	5	2146	160	160	0	Ja
Systemkomponenten müssen vorhanden sein wenn einer der o.g. Bausteine eingesetzt wird							
FC557	AI_KERNEL	-	512	-	-	-	Nein
FC558	AO_KERNEL	-	250	-	-	-	Nein
FC559	SI_KERNEL	-	596	-	-	-	Nein
	**Gesamt Grundinstallation*	*0*	***1358***			0	

*Erforderlich bei Verwendung von SCHEDULE oder CLOCK_TIMER

Funktionsbausteine Sm@rtLib für Schrittketten (Grundfunktionen)

Die Bausteine müssen nur im Bausteinordner vorhanden sein, wenn sie im Programm verwendet werden.

Nr	Bausteintyp	Proz-var.	Arbeitsspei-cher FB	Arbeitsspeicher Instanz		Anzahl Mel-debst	Option
FB582	FW_CONTROL	33	5788	266	395	1	Nein
	Fahrweise 2 Lauf Schritte	0	1500	200	200	0	Ja
	Fahrweise 5 Lauf Schritte	0	1900	200	200	0	Ja
	Fahrweise 10 Lauf Schritte	0	2600	200	200	0	Ja
Systemkomponenten müssen vorhanden sein wenn einer der o.g. Bausteine eingesetzt wird							
FB594	CS TIME	-	1170	-		-	Nein
FB595	STEP_MAN	-	204	-		-	Nein
FB582	SGRAPH	-	832	-		-	Nein
FC583	BCH_CTRL	-	1284	-		-	Nein
	**Gesamt Systembaus-teine*	*0*	***3490***			*0*	

4 Einrichten des Projekts

4.1 Vorraussetzungen

Für die Erstellung eines Projektes mit Sm@rtLib muss die Engineeringumgebung für STEP7, WinCC bzw. PCS7 eingerichtet sein (siehe „Software Vorraussetzungen“).

- Neues Projekt bzw. AS ist angelegt
- S7-AS Hardware ist eingefügt und konfiguriert
- WinCC OS ist angelegt (bei Visualisierung mit WinCC)
- WinCCFlexible Gerät ist eingefügt (bei Visualisierung mit WinCCflexible)
- Kommunikationsverbindung zwischen AS und OS sind projektiert
- Das Sm@rtLib Setup muss ausgeführt worden sein

Sm@rtLib kann in „normalen“ Projekten oder Multiprojekten verwendet werden. Bei PCS7 kann optional die technologische Hierarchie verwendet werden.
Die Kommunikation zwischen AS und OS ist über Ethernet, Profibus oder MPI möglich.

4.2 Einfügen der Bausteine und Pläne in das S7-Projekt

4.2.1 Verbindungsname

Nach der Konfiguration der S7-Hardware befindet sich unter der CPU der Ordner „S7-Programm“. Geben Sie diesem Ordner einen aussagekräftigen Namen (z.B. Teilanlagenbezeichnung), da dieser Name in WinCC als Bestandteil des Variablennamens verwendet wird. Der Name muss im Projekt eindeutig sein!

4.2.2 Sm@rtLib Bausteine undTextlibrary einfügen

Öffnen Sie im Simatic-Manager die erforderliche Bibliothek:

- **SLib3Cpl** für SIMATIC **S7-300**
- **SLib4Cpl** für SIMATIC **S7-400**

Objektname	Symbolischer Name	Typ	Größe	Autor	Änderungsdatum
Sources	---	Quellordner	---		14.03.2007 15:21:47
Blocks	---	Bausteinordner offline	---		28.03.2007 16:22:09
Charts		Planordner	1953...		23.03.2007 12:19:24
Text libraries	---	Textbibliotheksordner	---	---	02.02.2005 07:06:58
Symbols	---	Symboltabelle	22088		28.03.2007 16:22:10

In der Bibliothek befinden sich folgende Elemente:

Sources: Eine Quelle mit einem Beispiel zu einer Fahrweise
Blocks: Enthält die Sm@rtLib S7-Bausteine
Charts: Enthält Musterpläne für CFC
Text libraries: Textbibliotheken für die Meldegenerierung

Symbolik einfügen
Kopieren Sie die Symbolik in Ihr Projekt.
Achtung!
Sollten sich bereits Symboleinträge in dem Projekt befinden, dann werden diese überschrieben! In diesem Fall muss die Symbolik geöffnet werden und die Symbole mit Kopieren/Einfügen in das Projekt kopiert werden.

STEP7-Bausteine kopieren

Kopieren Sie alle Bausteine aus dem Ordner „Blocks“ in das Projekt.

Der benötigte Arbeitsspeicher für die Sm@rtLib Bausteine beträgt ca. 80 kB. Es wird empfohlen auch die nicht verwendeten Bausteine mit in das Projekt zu kopieren. So ist sichergestellt das auch alle benötigten Unterfunktionen verfügbar sind. Außerdem wird sichergestellt dass die Bausteine zur Version passen, was bei späteren Erweiterungen des Programms hilfreich ist.
Sollten Sie trotzdem Speicherplatz sparen wollen, dann können Sie die optionalen Bausteine aus dem Programmordner wieder löschen. Welche Bausteine optional sind können Sie in der Bausteinliste im Kapitel „Auswahl der CPU“ ersehen.

4.2.2.1 Textlibraries kopieren

Kopieren Sie den kompletten Ordner „Text libraries“ in das Projekt.
Sollte der Simatic-Manager dabei nach der Einstellung für die Meldenummern fragen, wählen Sie die Alternative:
Meldnummern immer CPU-weit eindeutig vergeben

4.2.3 Systemplan und Musterpläne

Wenn Sie CFC verwenden dann legen Sie in Ihrem Projekt den Ordner „Pläne“ an.

Die Pläne im Ordner „Charts“ der Sm@rtLib-Bibliothek dienen als Vorlage für das Projekt. Sie können die Pläne in Ihr Projekt kopieren und die Plannamen entsprechend anpassen.

Achtung!
Die Plannamen sind Bestandteil des WinCC-Variablennamens und sollten nach der Bilderstellung nicht mehr geändert werden!

Den Plan „SYSTEM“ müssen Sie auf jeden Fall in Ihr Projekt kopieren, da in diesem Plan die Sm@rtLib Systemfunktionen aufgerufen werden. Den Plan SYSTEMTIME müssen Sie nur verwenden, wenn Sie den SCHEDULER oder den CLOCK_TIMER verwenden wollen.

Objektname	Version	Typ	Autor	Änderungsdatum
A_INPUT	0.1	CFC		23.03.2007 12:17:47
A_OUTPUT	0.1	CFC		23.03.2007 12:17:47
ACTUATOR	0.1	CFC		23.03.2007 12:17:47
AEM_VALVE	0.1	CFC		23.03.2007 12:17:47
C_CTRL	0.1	CFC		23.03.2007 12:17:47
CLOCK_TIMER	0.1	CFC		23.03.2007 12:17:47
CONVERTER	0.1	CFC		23.03.2007 12:17:47
DIFF_INT	0.1	CFC		23.03.2007 12:17:47
DOS	0.1	CFC		23.03.2007 12:17:47
HT_CURVE	0.1	CFC		23.03.2007 12:17:47
LIMIT	0.1	CFC		23.03.2007 12:17:47
M_CONTACT	0.1	CFC		23.03.2007 12:17:47
MOTOR_1N	0.1	CFC		23.03.2007 12:17:47
MOTOR_2N	0.1	CFC		23.03.2007 12:17:47
MOTOR_LR	0.1	CFC		23.03.2007 12:17:47
RTM	0.1	CFC		23.03.2007 12:17:47
S_CTRL	0.1	CFC		23.03.2007 12:17:47
SCHEDULE	0.1	CFC		23.03.2007 12:17:47
SPLIT_RANGE	0.1	CFC		23.03.2007 12:17:47
SYSTEM	0.1	CFC		23.03.2007 12:17:47
SYSTEMTIME	0.1	CFC		23.03.2007 12:19:21
VALVE	0.1	CFC		23.03.2007 12:17:47

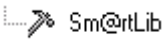
Achten Sie darauf, dass die Funktionsbausteine SYSTEM und OM_EVENT_MSG immer an erster Stelle der OBs eingebaut sind in denen der Baustein aufgerufen wird (z.B. OB35, OB100, OB1)

Für die STEP7 Umgebung sind nun alle Komponenten eingefügt. Sie können nun das Steuerungsprogramm erstellen oder mit der Einrichtung des Systems fortfahren. Legen Sie für den OS-Transfer mindestens eine Messstelle an und übersetzen Sie die Pläne im CFC Editor.

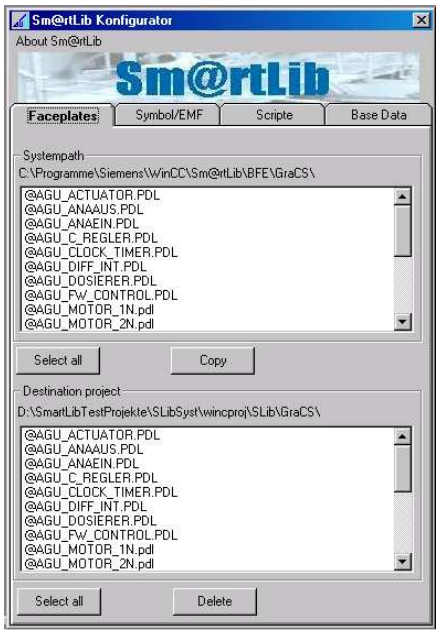
4.3 WinCC OS konfigurieren

Diese folgenden Einstellungen müssen Sie an einem neu erzeugten Projekt vornehmen oder wenn Sie Sm@rtLib das erste Mal in einer WinCC-OS verwenden.

4.3.1 Konfigurator

Öffnen Sie den WinCCExplorer. Im Programmmenü befindet sich der Eintrag 

. Über die rechte Maustaste wählen Sie Projektkonfiguration.
Es öffnet sich der Sm@rtLib Konfigurator. Der Konfigurator kopiert die erforderlichen Komponenten in das Projekt und legt ggf. fehlende Unterverzeichnisse an.



Faceplates:	Kopieren der PDL für die Sm@rtLib-Faceplates in das Projekt
Symbole:	Kopieren der PDL- und EMF-Dateien für die Bildsymbole in das Projekt
Scripte:	Kopieren der VisualBasic-Scripte für Sm@rtLib in das Projekt
BaseData:	Kopieren der modifizierten Sammelanzeige in das WinCC Systemverzeichnis und Sichern der Original Siemens Sammelanzeige

Um die Daten in das Projekt zu kopieren klicken Sie jeweils in auf den Button „Alle wählen“ und anschließen auf „Kopieren“.
Achtung: Im Projekt vorhandene Dateien werden ohne Warnung überschrieben!

Durch das Kopieren von BaseData werden die von WinCC installierten Sammelanzeigeneinstellungen überschrieben. Die Original Siemens Sammelanzeigen werden unter dem WinCC-Installationspfad im Verzeichnis ***\options\pdl\base_Data_Pool\BaseDataSiemensORI*** gesichert.
Wenn Sie die Original Siemens Installationsdateien wiederherstellen wollen, dann kopieren Sie die Dateien aus diesem Verzeichnis in das Verzeichnis ***\options\pdl\base_Data_Pool*** zurück.
Um die modifizierte Sammelanzeige zu verwenden müssen Sie anschließend den OS-Projekteditor durchlaufen lassen. Diese kopiert die Sammelanzeige entsprechend der Bildschirmkonfiguration in das Projekt.

Über den Konfigurator können die Sm@rtLib Komponenten ebenfalls aktualisiert werden, wenn Sie eine neue Version von Sm@rtLib verwenden.

4.3.2 Alarm-Logging anpassen

Bevor Sie die Meldegruppen im Alarm-Logging anpassen können müssen Sie mit dem OS-Projekteditor für WinCC das Alarm-Logging eingerichtet haben.

Damit die Farbgebung für Meldklassen mit der Farbgebung im Alarm-Logging übereinstimmt müssen einige Meldklassen angepasst werden.

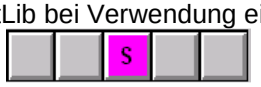
Meldeklasse	Unterpunkt	SmartLib relevant	Textfarbe	Hintergrund	WinCC default	Bemerkung
Alarm	Alarm High	400	weiß		x	
	Alarm Low	400	weiß		x	
Warnung	Warning High	400	schwarz		x	
	Warning Low	400	schwarz		x	
Toleranz	Tolerance High	-	weiß		x	
	Tolerance Low	-	weiß		x	
AS Leittechnik Meldung	System	300/400	schwarz	magenta		Störung
	Fehler	400	schwarz	cyan		Verriegelungsstörung
OS Leittechnik Meldung	System	-	gelb	schwarz	x	
	Fehler	-	gelb	schwarz	x	
Vorbeugende Wartung	Wartung	-	gelb	schwarz	x	
Prozessmeldung	Prozessmeldung	-	weiß	violett	x	
Betriebsmeldung	Betriebsmeldung	300	schwarz	hellgrau	x	
Bedienanforderung	Bedienanforderung	-	weiß	blau	x	
Bedienmeldung	Bedienmeldung	300/400	weiß	violett	x	
Status Meldung	Status-AS	-	schwarz	hellgrau	x	
	Status-OS	-	schwarz	hellgrau	x	
System quittierpflichtig	Leittechnik	-	schwarz	hellgrau	x	
	Systemmeldungen	-	schwarz	hellgrau	x	
System, nicht quittierpflichtig	Leittechnik	-	schwarz	hellgrau	x	
	Bedienmeldungen	-	schwarz	hellgrau	x	

4.3.3 Sammelanzeigen anpassen

Ab der V4.1.0 wird die Sammelanzeige durch den Sm@rtLib-Projektkonfigurator automatisch richtig eingestellt. Sollten Sie den Projektkonfigurator nicht verwenden oder abweichende Einstellungen vornehmen wollen, werden die Einstellungen hier beschrieben. Wenn Sie die Sammelanzeige über den Projektkonfigurator kopiert haben können Sie diesen Punkt übergehen.

Zur Einstellung der Sammelanzeigen für das WinCC-Basic-Process-Control auf die Sm@rtLib Farbeinstellungen müssen Sie die WinCC-Standard Sammelanzeige im Graphics Designer anpassen.

Sammelanzeige Sm@rtLib bei Verwendung einer S7-300



Sammelanzeige Sm@rtLib bei Verwendung einer S7-400

Die Sammelanzeigen für die Bereichsübersicht liegen im GraCS-Verzeichnis. Dateiname: @AreaButtons.
Die Farben können in den Objekteigenschaften der Sammelanzeige unter *Meldearten* angepasst werden. Folgende Einstellungen entsprechen den Sm@rtLib Farbfestlegungen:

Attribut	**Statik**					
Meldearten	**Alarm High /Alarm Low**	**Warning High / Warning Low**	**Toleranz High /Toleranz Low**	**AS Leittechnik Störung**	**AS Leittechnik Fehler**	**OS Leittechnik Störung**
Anzeigetext	A	W	T	S	V	S
Gekommen-Schrift blinken	nein	nein	nein	nein	nein	ja
Gekommen-Schriftfarbe Ein	weiß	schwarz	weiß	schwarz	schwarz	gelb
Gekommen-Schriftfarbe Aus	weiß	schwarz	weiß	schwarz	schwarz	schwarz
Gekommen-Hintergund blinken	ja	ja	nein	ja	ja	nein
Gekommen-Hintergrundfarbe Ein	rot	gelb	hellblau	magenta	cyan	schwarz
Gekommen-Hintergrundfarbe Aus	grau	grau	hellblau	grau	grau	schwarz
Gekommen Quittiert-Schrift blinken	nein	nein	nein	nein	nein	nein
Gekommen Quittiert-Schriftfarbe Ein	weiß	schwarz	weiß	schwarz	schwarz	gelb
Gekommen Quittiert-Schriftfarbe Aus	weiß	schwarz	weiß	schwarz	schwarz	gelb
Gekommen Quittiert-Hintergrund blinken	nein	nein	nein	nein	nein	nein
Gekommen Quittiert-Hintergrundfarbe Ein	rot	gelb	hellblau	magenta	cyan	schwarz
Gekommen Quittiert-Hintergrundfarbe Aus	rot	gelb	hellblau	magenta	cyan	schwarz
Gegangen Unquittiert-Schrift blinken	nein	nein	nein	nein	nein	ja
Gegangen Unquittiert-Schriftfarbe Ein	weiß	schwarz	weiß	schwarz	schwarz	gelb
Gegangen Unquittiert-Schriftfarbe Aus	weiß	schwarz	weiß	schwarz	schwarz	schwarz
Gegangen Unquittiert-Hintergrund blinken	ja	ja	nein	ja	ja	ja
Gegangen Unquittiert-Hintergrundfarbe Ein	rot	gelb	hellblau	magenta	cyan	schwarz
Gegangen Unquittiert-Hintergrundfarbe Aus	grau	grau	hellblau	grau	grau	schwarz

Nach Abschluss dieser Einstellungen sind alle erforderlichen Sm@rtLib-Daten im Projekt und die erforderlichen Voreinstellungen durchgeführt.

4.3.4 Bedienberechtigungen festlegen

Im UserAdministrator werden die Bedienberechtigungen für WinCC eingerichtet. Die hier aufgeführenten Berechtigungsstufen sind die SmartLib Defaulteinstellungen. Diese sind in den Bildsymbolen bereits so konfiguriert. Sie können die Einstellungen für Ihr Projekt frei konfigurieren. Dann müssen alllerdings in den Bildsymbolen mit Faceplateaufruf die Berechtigungsstufen ebenfalls angepasst werden.

Die folgende Bedienberechtigungstufen sind SmartLib-Default:

Benutzergruppen:

1. **Anlagenfahrer (vor Ort Personal der Produktion)**
   - Default User (wenn sonst keiner eingeloggt ist)
   - Reine Beobachtung der Anlage
   - Prozessbedienung
   - Höherwertige Prozessbedienung
   - Reportsystem
   - Archive bedienen
   - Grundfunktion einschalten
   - Fahrweisen der Grundfunktion umschalten

2. **Produktionsmeister**
   - Eingabe von Sollwerten (GF)
   - Eingabe von Sollwerten (GFE)
   - Handebene freigeben
   - Regler ausschalten
   - Änderung von Toleranzgrenzen
   - Änderung von Warngrenzen

3. **Instandhaltungshandwerker PCS / Entstördienst**
   - Projektierung und Parametrierung des Systems
     - Änderung von Grenzwerten
     - Änderung von Reglerparametern

4. **Ingenieur PCS**
   - Freier Zugang zum System

5. **Superuser (Auftraggeber)**
   - Freier Zugang zum System

Stufe Zugangsberechtigung

10 keine Bedienberechtigung
16 Grundfunktion einschalten
18 Fahrweisen der Grundfunktion umschalten
20 Eingabe von Sollwerten (GF)
25 Eingabe von Sollwerten (GFE)
30 Handebene freigeben
31 Regler ausschalten
38 Änderung von Toleranzgrenzen
39 Änderung von Warngrenzen
40 Änderung von Grenzwerten
41 Änderung von Reglerparametern
50 Inbetriebnahmeistwert (PLT-Taste)

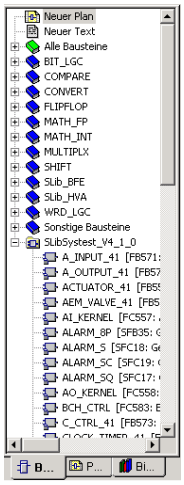
5 Projektieren eine Messstelle

In diesem Kapitel werden die Projektierungsschritte beim Anlegen einer Messstelle am Beispiel eines Reglers beschrieben. Das Zielsystem ist eine S7-300. Die Projektierung erfolgt in CFC-Technik

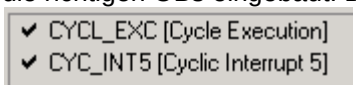
Voraussetzungen: Das Grundprojekt mit AS und OS wurde angelegt und die unter „Einrichten eines Projektes“ beschriebenen Schritte wurden durchgeführt.

5.1 Anlegen und konfigurieren der Messstelle in CFC

- Erzeugen Sie im Simatic-Manager im Planordner des Projekts einen neuen CFC-Plan *Planordner>rechte Maus>neues Objekt einfügen>CFC*
- Ändern Sie den Namen des CFC-Plans in „V100_CA10_T01“.
- Öffnen Sie den Plan durch Doppelklick auf den CFC-Plan
- Wenn die Katalogansicht noch nicht geöffnet ist, können Sie diese mit *Ctlr-K* einschalten.
- Durch Anwahl des linken Karteireiter *Bausteine* werden die S7-Bausteine im Programmordner angezeigt



- Wählen Sie aus dem Katalog den Baustein C_CTRL_41 (FB573) und ziehen Sie den Baustein in den CFC-Plan
- Der C_CTRL Baustein muss im OB1 und OB35 aufgerufen werden. Der Baustein wird automatisch in die richtigen OBs eingebaut. Durch Doppelklick auf den Plan können Sie den Einbau kontrollieren.



CYCL_EXC = OB1, CYC_INT5 = OB35

- Doppelklick auf den Eingang Name_ID öffnet ein Dialogfeld. In das Feld *Text0:* tragen Sie die Kurzbezeichnung der Messstelle ein. Hier: V100_CA10_T01. Im Feld *Text2:* Kann ein Zusatztext eingetragen werden z.B. „Temperatur RA1“
- Zur Konfiguration des Reglers können die relevanten Parameter sichtbar geschaltet werden. Doppelklick auf den Baustein>Karteireiter *Anschlüsse*
  Entfernen Sie die Häkchen in der Spalte unsichtbar bei folgenden Parametern:

PV_HR / PV_LR / SP_OP_HL / SP_OP_LL / MV_HR / MV_LR / ADR_AI / ADR_AO

- Stellen Sie die erforderlichen Parameter jeweils durch Doppelklick auf den Eingang ein.
  PV_HR: 500, PV_LR: 0 Messbereich der Temperaturmessung 0-500℃
  SP_OP_HL: 300, SP_OP_LL: 100 Eingabegrenzen für den Sollwert 100-300℃
  MV_HR: 100, MV_LR: 0 Bereich für das Stellglied 0-100%
  ADR_AI: 256 PEW-Adresse des Analog-Eingangs (siehe HW-Konfig.)
  ADR_AO: 312 PAW-Adresse des Analog-Ausgangs für das Stellglied

- Übersetzen Sie die CFC-Pläne als Programm

- Übersetzen Sie die OS im Simatic-Manager

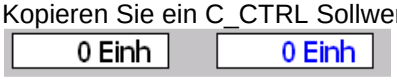
5.2 Anlegen und konfigurieren der Messstelle in WinCC

Durch das OS-Übersetzen sind in WinCC die erforderlichen Meldungen und Prozessvariablen angelegt worden. Für die Visualisierung des Reglers muss nun ein Symbol in ein Bild eingebaut werden.

- Öffnen Sie das entsprechende Bild im Graphics-Designer.

- Öffnen Sie ebenfalls das Bild *@AGU_SmartLibUDO_FB_Symbole*

- Kopieren Sie ein C_CTRL Sollwert und Istwertfeld aus dem Bild in Ihr Projektbild.



- Wählen Sie das Sollwertfeld an und öffnen Sie über das Menü Sm@rtLib>Symboleigenschaften den Konfigurationsdialog.

- Stellen Sie den Faceplatetyp auf C_CTRL

- Rufen Sie über die über "…" –Taste den Variablendialog und wählen Sie die Messstelle V100_CA10_T01/1 aus. Bestätigen Sie mit OK. Der Instanzname wird nun eingetragen.

- Tragen Sie einen Tooltip-Text ein

- Wählen Sie den Punkt Sammelrelevant an. Dadurch werden Störungen der Messstelle in der Sammelanzeige in der Bereichsübersicht angezeigt.

- Öffnen Sie über die Taste Berechtigungen den Dialog für die Bedienberechtigungen und tragen Sie die entsprechende Berechtigungsstufe für die Bedienungen ein oder wählen Sie die Berechtigungsstufe mit der Taste „…" aus dem UserAdministrator aus.

- Wiederholen Sie den Vorgang mit dem Istwertfeld.

Damit ist die Projektierung für die Steuerung und die Visualisierung abgeschlossen. Die Projektierung kann jetzt in die Steuerung bzw. Visualisierung übertragen werden. Das Faceplate wird in der WinCC-Runtime durch Anklicken des Reglersoll- oder Istwertes geöffnet.

6 Systembausteine:

6.1 SYSTEM

Der Baustein ruft die Systemfunktion SFC 6 (RD_SYSINFO) auf. Die Systemfunktion ist in einer CPU enthalten und braucht nicht geladen zu werden.

Der Baustein benötigt einen UserDefinedTyp UDT 6. Die Typdefinition liegt ebenfalls als Quelle vor und muss in der Symbolik eingetragen sein.

Der Systemplan, in den die Systembausteine implementiert werden, wird in der Komponentensicht im Planbehälter angelegt. Somit erscheint dieser Plan nicht in der technologischen Hierarchie.
Der Systemplan sollte aus dem Planbehälter der Bibliothek in den Planbehälter des Projektes kopiert werden. Damit sind alle Einstellungen an den Plänen vorgenommen.

Der Baustein BedBetrMeld wird im Systembaustein (AWL bzw. CFC Plan) aufgerufen (siehe Musterprojekt). Er ist in folgende Ablaufebenen einzubauen:

Ablaufebenen:

OB1	Zyklisches Programm, FZ freier Zyklus (CYC_EXC)
OB35	Weckalarm-OB, Weckalarm WAL5 (CYC_INT5)
OB100	Anlauf-OB, Neustart (COMPLETE RESTART)
OB122	Peripheriezugriffsfehler

Der Baustein kann in beliebigen weiteren Fehler-OB´s aufgerufen werden!

6.2 MSG_KERNEL

Dieser Baustein bildet die Schnittstelle der technologischen Bausteine zum Meldesystem des Standards. Der Baustein wird von jedem meldefähigen Baustein im Standard aufgerufen.
Er hat die Aufgabe, die Störung unmittelbar zu melden und den Störgrund in Form einer Betriebsmeldung in den Betriebsmeldepuffer EVENT_MSG_BUFFER einzutragen. Die Eintragung erfolgt mit der AS-Meldenummer (EV_ID) und der Störgrund-ID. Die Bedienmeldungen werden von diesem Baustein in den Bedienmeldungspuffer OM_MSG_BUFFER mit AS-Meldungsnummer und Bedienungs-ID eingetragen. Der Baustein unterstützt 8 Störmeldungen und eine Bedienmeldung.

6.3 OM_EVENT_MSG

Dieser Baustein bildet die Schnittstelle des Meldesystems des Standards zu den OSen (Bedienstationen). Der Baustein sendet die in den Betriebsmeldungspuffer BETR_MELPU und in den Bedienmeldungspuffer BEDI_MELPU eingetragenen Meldungen an die Bedienstationen und verwaltet die Meldepuffer.

Der Systemplan, in den die Systembausteine implementiert werden, wird in der Komponentensicht im Planbehälter angelegt. Somit erscheint dieser Plan nicht in der technologischen Hierarchie.
Der Systemplan sollte aus dem Planbehälter der Bibliothek in den Planbehälter des Projektes kopiert werden. Damit sind alle Einstellungen an den Plänen vorgenommen.

Der Baustein BedBetrMeld wird im Systembaustein (AWL bzw. CFC Plan) aufgerufen (siehe Musterprojekt). Er ist in folgende Ablaufebenen einzubauen:

Ablaufebenen:

OB1 Zyklisches Programm, FZ freier Zyklus (CYC_EXC)

Der Baustein darf ausschließlich im OB1 aufgerufen werden!

6.4 AI_KERNEL

Der Baustein wird intern vom Standard genutzt, um Analogwerte von der Peripherie einzulesen.

Dieser Baustein wird nur intern verwendet!

6.5 AO_KERNEL

Der Baustein wird intern vom Standard genutzt, um Analogwerte zu schreiben.

Dieser Baustein wird nur intern verwendet!

6.6 CONDITIONS

Der Baustein wird intern vom Standard genutzt um die Randbedingungen (STOP, ILOC, TOEB, ENAB) zu verarbeiten.

6.7 Besonderheit Systembausteine in einem AWL-Programm

Bei Verwendung des CFC-Editors werden die benötigten Systembausteine automatisch in die richtigen OBs eingebaut. Für den Einsatz der SmartLib in einem AWL oder FUP-Programm müssen die erforderlichen Bausteine vom Programmierer in die entsprechenden OBs eingebaut werden.

Dies sind die Bausteine:

FB551	SYSTEM	Einbau in OB 1, 35, 100, 122	
FB563	OM_EVENT_MSG	Einbau in OB 1	
FB580	DATETIME	Einbau in OB35	(Nur wenn SCHEDULER oder CLOCK_TIMER benötigt wird)

Die Auswahl der Datenbausteine für die System-FBs kann beliebig erfolgen. Hier wurden beispielhaft die DB10, DB12 und DB13 gewählt.

Wichtig ist, das der Aufruf des FB551 in allen OBs mit dem gleichen Instanz-DB erfolgt!

OB1 Programmsequenz

```
CALL  "SYSTEM" , DB10
 PULSE_T_1:=
 PULSE_T_2:=
 TASK_T   :=


CALL  "OM_EVENT_MSG" , DB12
```

OB35 Programmsequenz

```
CALL  "SYSTEM" , DB10
 PULSE_T_1:=
 PULSE_T_2:=
 TASK_T   :=


CALL  "DATETIME_41" , DB13
 T_CONNECT:=
 SEC      :=
```

OB100 Programmsequenz

```
CALL  "SYSTEM" , DB10
 PULSE_T_1:=
 PULSE_T_2:=
 TASK_T   :=
```

OB122 Programmsequenz

```
CALL  "SYSTEM" , DB10
 PULSE_T_1:=
 PULSE_T_2:=
 TASK_T   :=
```

6.8 Systembausteine in einem CFC-Programm

In den System CFC Plan sind die drei Systembausteine integriert. Der DATETIME Baustein hat Verbindungen zu allen Bausteinen im Programm, die Datum und Uhrzeit abhängig sind.

7 Meldesystem

Sm@rtLib besitzt verschiedene Meldeklassen. Dabei beziehen sich die Meldeklassen auf Meldungen, die durch die Sm@rtLib-Bausteine generiert werden. Systemspezifische Meldungen, wie z.B. der Ausfall eines Servers oder der Ausfall einer CPU werden im Sm@rtLib-Standard nicht betrachtet. Diese sind losgelöst von Sm@rtLib in Abhängigkeit des jeweiligen eingesetzten Prozessleitsystems zu betrachten.

7.1 Meldefähigkeit mit dem Alarm Baustein der CPU

Der Einsatz der Funktionsbausteine in einer S7-300 Steuerung und S7-400 Steuerung mit voller Meldefähigkeit ist möglich, da für die Meldungen der Alarm_S Baustein verwendet wurde.
In den Steuerungen können gleichzeitig (Meldeschauer) jedoch immer nur so viele Meldungen generiert werden wie meldefähige Bausteine in der Steuerung vorhanden sind.

Jeder Sm@rtLib Funktionsbaustein der eine Meldung in WinCC generiert, nutzt einen meldefähigen Baustein der Steuerung.

Es gilt folgende Regel: **MBx = MB - 2**

MBx = Anzahl der nutzbaren gleichzeitigen Meldungen im Programm
MB = Anzahl meldefähiger Baustein in der Steuerung

Ungeachtet der Begrenzung der meldefähigen Bausteine (MBx) können Sie in ihr Projekt mehr Sm@rtLib Funktionsbausteine integrieren als die Begrenzung (MBx) vorgibt.
Beachten Sie nur, das die gleichzeitigen Meldungen begrenzt sind. Im Falle einer Meldeschauer von mehr als MBx Meldungen würde jede weitere Meldung nicht in WinCC eingetragen.

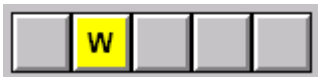
7.2 Meldesystem – S7-400 Steuerung

Sm@rtLib kann in S7-400 Steuerungen eingesetzt werden und nutzt die Alarm 8P Funktionen für die verschiedenen Meldeklassen.

Die folgenden Meldeklassen werden generiert.

quittierpflichtig

Alarm: Es ist ein Fehler aufgetreten, der einen Schutz des Verfahrens oder der Anlage erfordert und in der Regel eine Abschaltung bewirkt. Beispiel: Grenzwertverletzung, Einlauf- oder Nachlaufstörung



quittierpflichtig

Warnung: Es werden Hinweise angezeigt, die zu einem kritischen Zustand führen können. Ferner werden in dieser Klasse Hinweise auf Wartungsarbeiten gegeben. Beispiel: Verletzung von Warngrenzen, Zustand Sicherheitsschalter

quittierpflichtig

Störung: In dieser Klasse werden alle Baugruppenfehler, Messwertfehler, Bimetallstörung und Rückmeldestörungen vereint.

quittierpflichtig

Verriegelung: In dieser Klasse werden alle Verriegelungsstörungen der Grundfunktionselemente (GFE) und Grundfunktionen (GF) vereint.

quittierpflichtig

Bedienanforderung: Es ist eine Eingabe oder eine Bestätigung durch den Bediener notwendig.

7.3 Meldesystem – S7-300 Steuerung

Die Steuerung S7-300 unterstützt die AlarmS Funktionen und kann nur eine Meldeklasse generieren. Sofern eine Störung (Alarm, Warnung, Störung, o.a.) in einem Sm@rtLib Baustein auftritt, wird eine Sammelstörung generiert und an das Meldesystem gesendet. Die Differenzierung des Störgrundes kann im Faceplate erfolgen.

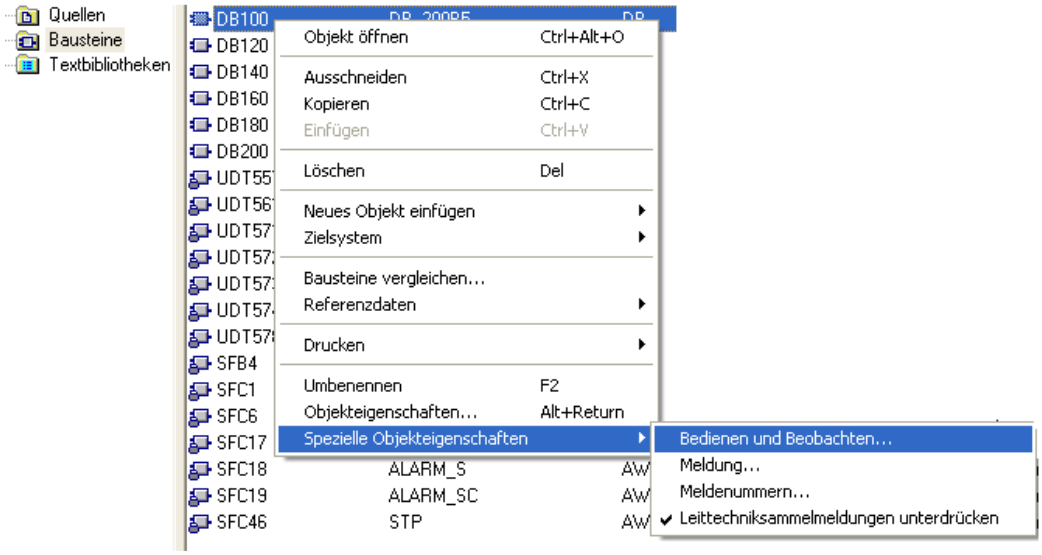
quittierpflichtig

Sammelstörung: In dieser Klasse werden alle Alarme-, Warnungen-, Verriegelungsstörungen zusammengefasst.

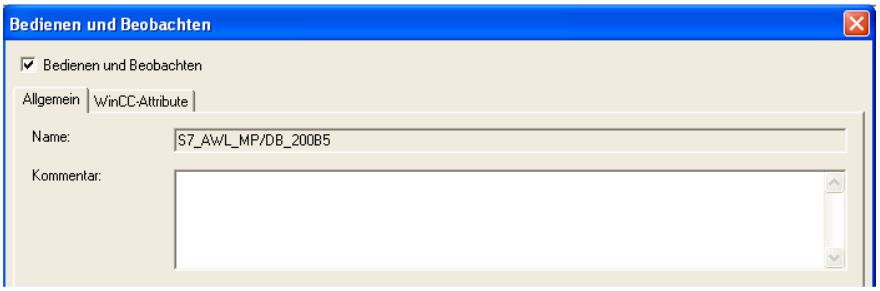
7.3.1 AWL Programme

Innerhalb von CFC-Plan Programmen erfolgen die folgenden Einstellungen automatisch. In AWL müssen folgende Einstellungen durch den Programmierer vorgenommen werden.

Instanz Datenbausteine müssen mit dem Attribut „Bedienen und Beobachten“ gekennzeichnet werden. Die so gekennzeichneten Bausteine erzeugen beim „Übersetzen“ ihre Variablen in WinCC/PCS7 automatisch.

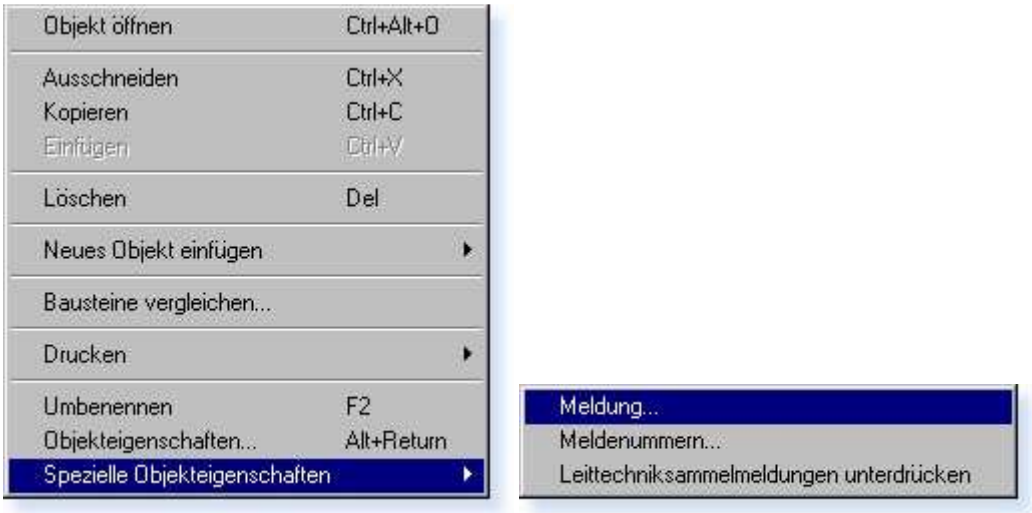


Folgende Untermaske erscheint. Der Haken „Bedienen und Beobachten“ muss gesetzt werden.

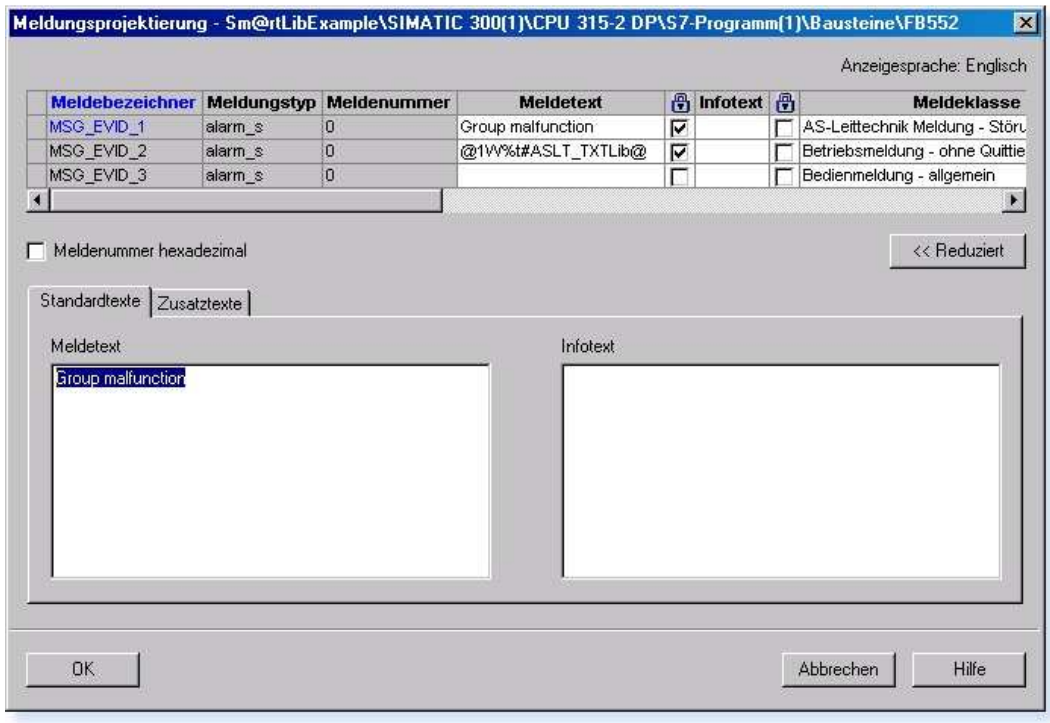


7.4 Meldesystem einstellen

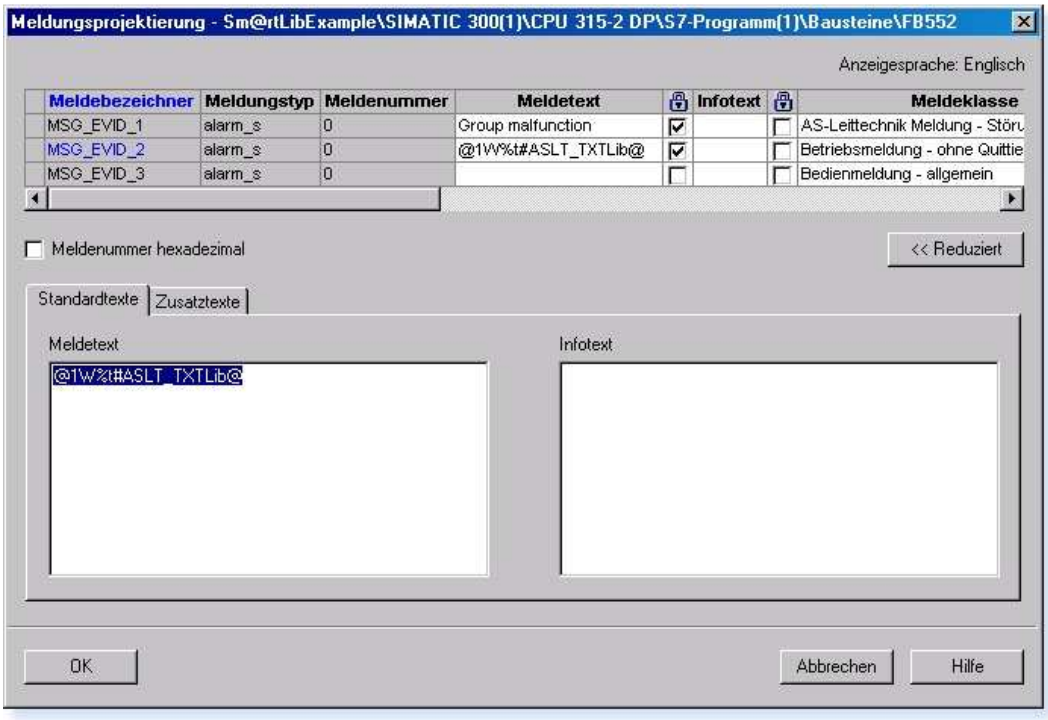
Jeder meldefähige Baustein hat eine Einstellung der Meldeattribute. Über die Objekteigenschaften des Bausteins kommt man zu den Speziellen Objekteigenschaften den Meldungen des Bausteins. Für alle Bausteine gelten die gleichen Einstellungen.
Diese Einstellungen sind bereits für die Sm@rtLib Bausteinen enthalten.



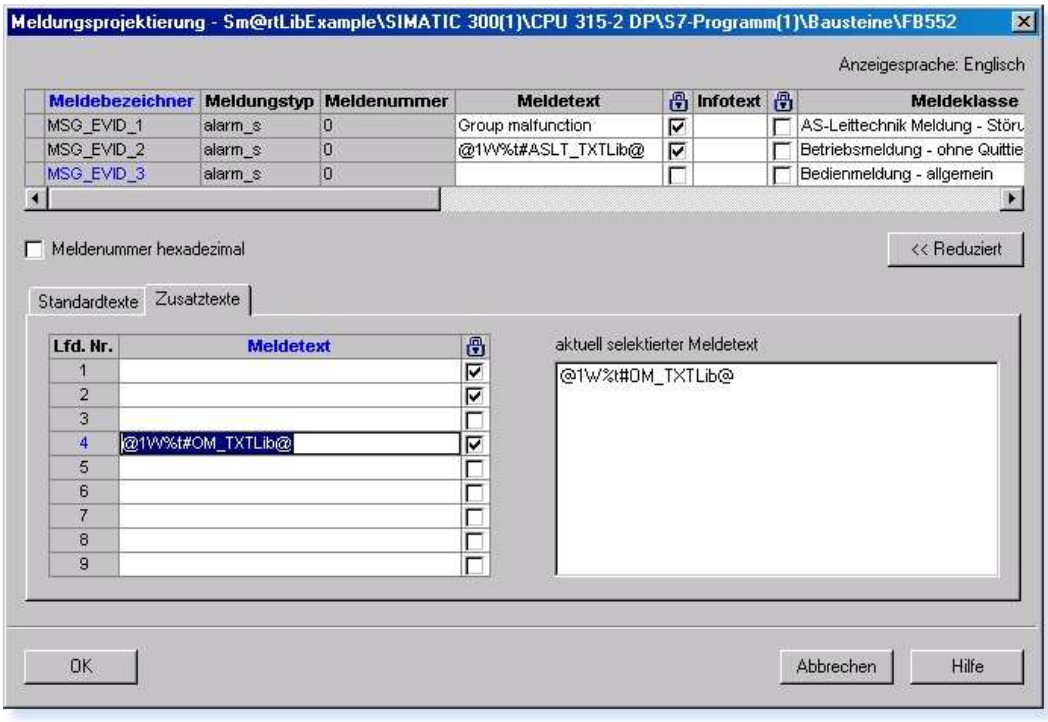
Einstellungen der Sammelstörmeldung MSG_EVID_1:



Einstellungen der Störgrundmeldung MSG_EVID_2:



Einstellung der Bedienmeldung MSG_EVID_3:



8 Quittierungskonzept

8.1 Quittierung Status

Der Status einer gekommenen Störung, die als blinkende Zustandsanzeige visualisiert oder als Meldung in die Neuseite eingetragen wird, wird mit der Taste im Karteireiter des Faceplate (oder der Quittiertaste im Übersichtsbereich oder in der Neuseite) zur Kenntnis genommen und quittiert. Die blinkende Zustandsanzeige wechselt in Ruhiglicht und die Meldung wird aus der Neuseite in die Altseite übertragen, solange die Störung noch ansteht. Der Status gilt jetzt als quittiert.

Durch diese Quittierung ist allerdings nur die Meldung quittiert worden. Damit z.B. ein Aktor nach Behebung der Störung nicht unkontrolliert wieder anläuft muss die Störung über die **QS** Taste im Faceplate quittiert werden und damit der Aktor wieder freigegeben werden. Anschließend muss der Aktor durch Handbedienung oder Flanke am AUTO_ON - Eingang wieder gestartet werden. Statt der QM-Taste kann der Baustein auch über den Ack-Eingang des Bausteins über das Programm quittiert werden.

9 WinCC Grundlagen – ab SmartLib V4

9.1 Festlegung der Symbole ab SmartLib V4.1.0

Für die Symboldarstellung in den Fließbildern gibt es zwei Varianten.

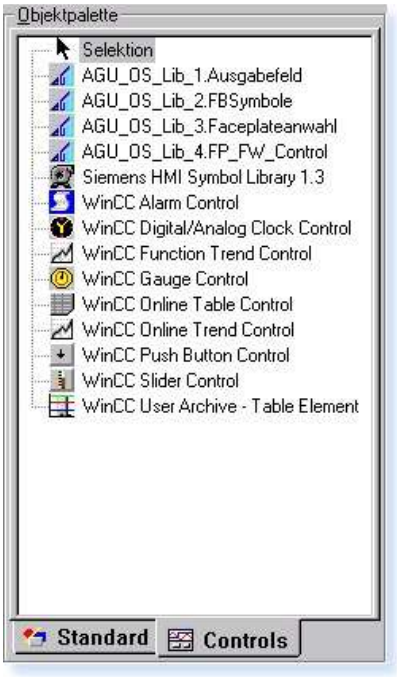
- ActiveX Controls
- UDO (**U**ser **d**efined **o**bjects)

Wegen der größeren Flexibilität in der Darstellung und der Möglichkeit zur projektspezifischen Anpassung ist bei neuen Projekten die Verwendung von UDO zu empfehlen!

9.2 Verwendung von ActiveX-Controls

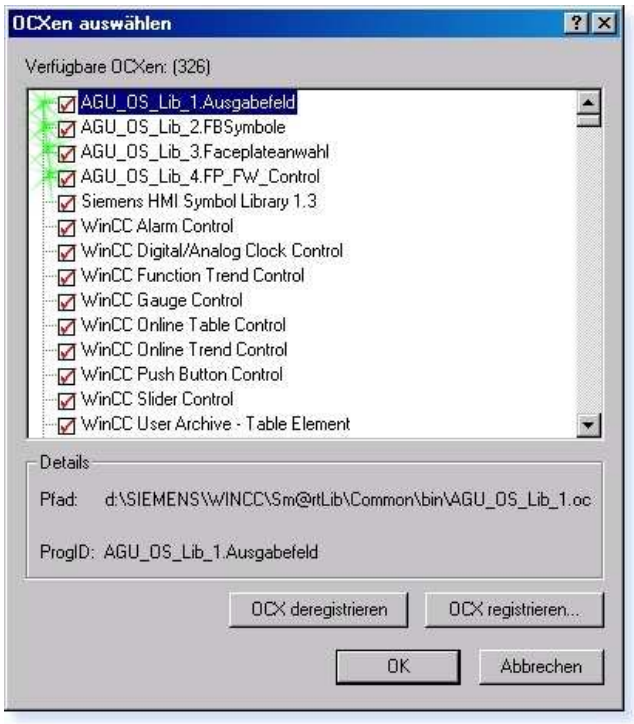
Die ActiveX-Controls können im Graphics-Designer verwendet werden, wenn Sie in der Objektpalette sichtbar geschaltet worden sind.

Wenn Sie innerhalb der Objektpalette (roter Stern) mit der rechten Maustaste klicken, erscheint ein Button "Hinzufügen/Entfernen". Über diesen Button kommen Sie in die Maske der COM-Objekte des Betriebssystems.



Drücken sie nach der Auswahl auf

Auswahl der OCX Elemente



Wählen Sie in der Auflistung die Objekte "AGU_OS_Lib_1", "AGU_OS_Lib_2" und "AGU_OS_Lib_3" aus.

Bestätigen Sie diese Maske mit "OK" und die Controls stehen Ihnen in der Objektpalette zur Verfügung.

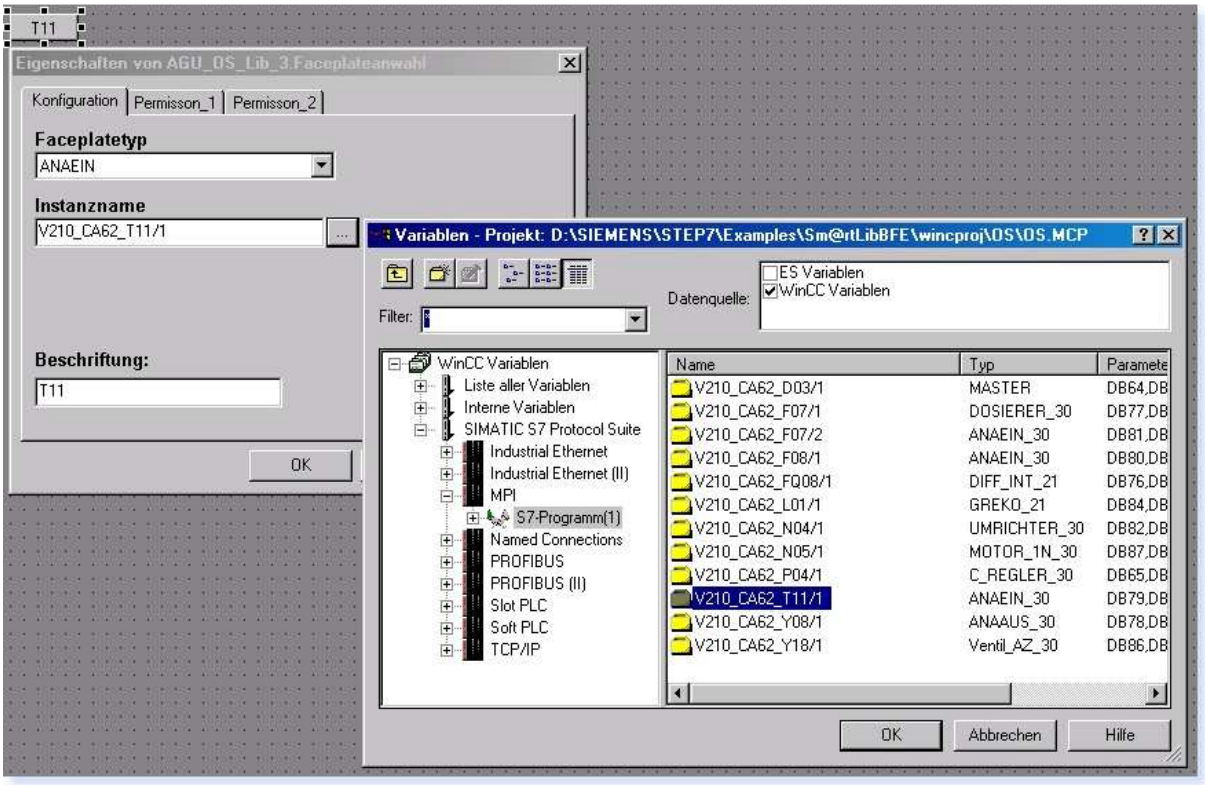
9.3 Anwahlbutton für den Faceplate Aufruf für WinCC

Im Sm@rtLib gibt es zum Aufruf der Faceplates einen Anwahlbutton, der die Aufgabe hat, alle instanzspezifischen Information an das Faceplate zu übergeben und das Faceplate im Basic Process Control von WinCC aufzurufen.

Über einen Doppelklick in der Objektpalette/Controls auf das AGU_OS_Lib_3.Faceplateanwahl wird das Control ins Bild gelegt. Ein Doppelklick auf das Control im Bild öffnet den Parametrierdialog.

An der Auswahlliste "Faceplatetyp" kann der Faceplatetyp ausgewählt werden. Passend zum Faceplatetyp muss jetzt die Strukturvariable von WinCC gewählt werden. Über einen Klick auf den Button am Eingabefeld "Instanzname" wird der Auswahldialog der Strukturvariablen von WinCC geöffnet. Hier wählen Sie die Strukturvariable und beenden den Dialog mit OK.

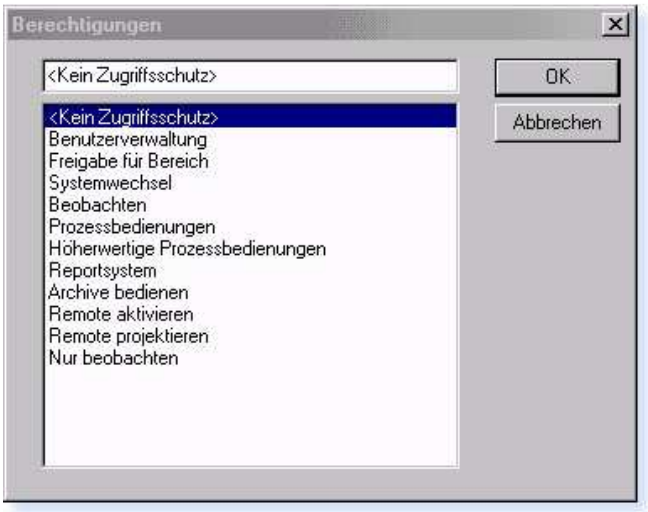
Die Beschriftung des Anwahlbuttons tragen Sie in das Feld "Beschriftung" ein.



In der zweiten und dritten Eigenschaftsmaske "Permisson_1" und "Permisson_2" werden die Bedienberechtigungen der Faceplates eingestellt. Abhängig vom gewählten Faceplate werden die Zuordnungen der Bedienberechtigungen geändert. Die Beschreibung legt fest, worauf die Berechtigung im Faceplate wirkt. Die Berechtigung wird als Nummer und als Beschreibung angezeigt.
Über den Auswahlbutton an der Berechtigung wird der Auswahldialog der UserAdministration von WinCC geöffnet. Hier können Sie zwischen allen von Ihnen projektierten Berechtigungen wählen.

Einstellung der Bedienberechtigung auf spezielle Bedienungen.

Auswahl der Bedienberechtigung



Wenn Sie im Graphics-Designer den Anwahlbutton auswählen und anschliessend den Eigenschaftsdialog von WinCC wählen können Sie noch weitere Attribute des Controls ändern. Sie können die Schriftart der Caption und die Farbe des Buttons beeinflussen.

In der folgenden Tabelle sind alle Eigenschaften außer den Standardeigenschaften des Objektes aufgelistet:

Wert	**Wertbeschreibung**	**Eigenschafts-seite**
Instanzname	Variablenname der Strukturvariablen der Instanz, die vom OS-Übersetzer aus dem CFC-Plan generiert wurde.	1
Caption	Beschriftung des Anwahlbuttons.	1
Faceplatetyp	Auswahl des Faceplates welches geöffnet wird.	
Permission 1 -10	Bedienberechtigungsstufen (1,2, ... 10) welche an das Faceplate übergeben werden. Jede Berechtigung kann einer Bedienung am Faceplate zugeordnet sein. Für jede Instanz können die Berechtigungen neu vergeben werden.	2
Permission 11 -20	Bedienberechtigungsstufen (11,12, ... 20) welche an das Faceplate übergeben werden. Jede Berechtigung kann einer Bedienung am Faceplate zugeordnet sein. Für jede Instanz können die Berechtigungen neu vergeben werden.	3
Font	Einstellen der Schriftart der Caption.	-
BackColor	Hintergrundfarbe des Anwahlbuttons.	-

9.4 Fließbildsymbol für WinCC

In der Eigenschaftsseite "Bild" werden den einzelnen Stati der Instanzen Bilder zugeordnet.

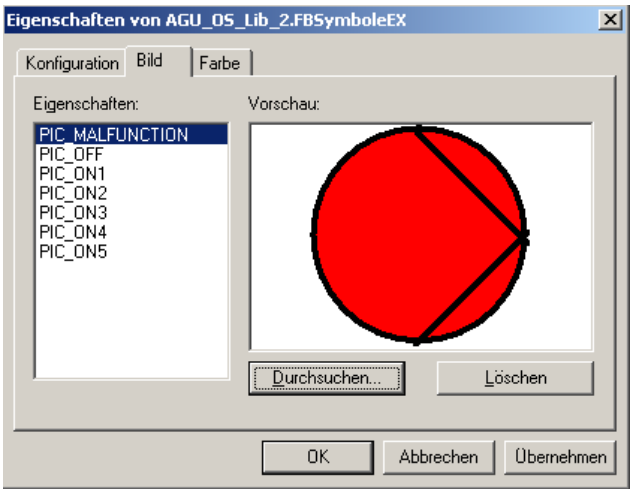
Bild/Instanztyp	Beschreibung	Ventil_AZ	MOTOR_1N	ANAAUS	UMRICHTER
PIC_AUS	Zustand AUS	x	x	x	x
PIC_BGF	Zustand Baugruppenfehler Analogwert	-	-	x	x
PIC_BIM	Zustand BI-Metallstörung	-	x	-	x
PIC_EIN	Zustand EIN	x	x	x	x
PIC_RCST	Zustand Rückmelde-störung	x	x	x	x
PIC_RS	Zustand Reparatur-schalter	-	x	-	x
PIC_VS	Zustand Verriegelungs-störung	x	x	x	x

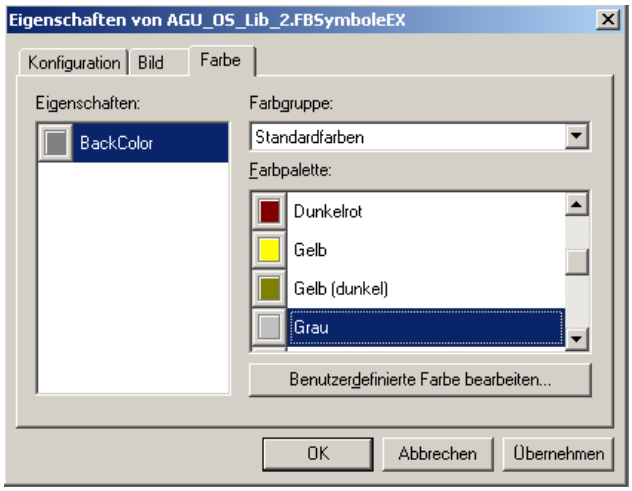
In der Eigenschaftsseite "Farbe" wird die Hintergrundfarbe des Objektes eingestellt. Sinnvoll ist hier die Farbe des WinCC-Bildes einzustellen.

Ein Aktorsymbol wird über den folgenden Dialog konfiguriert

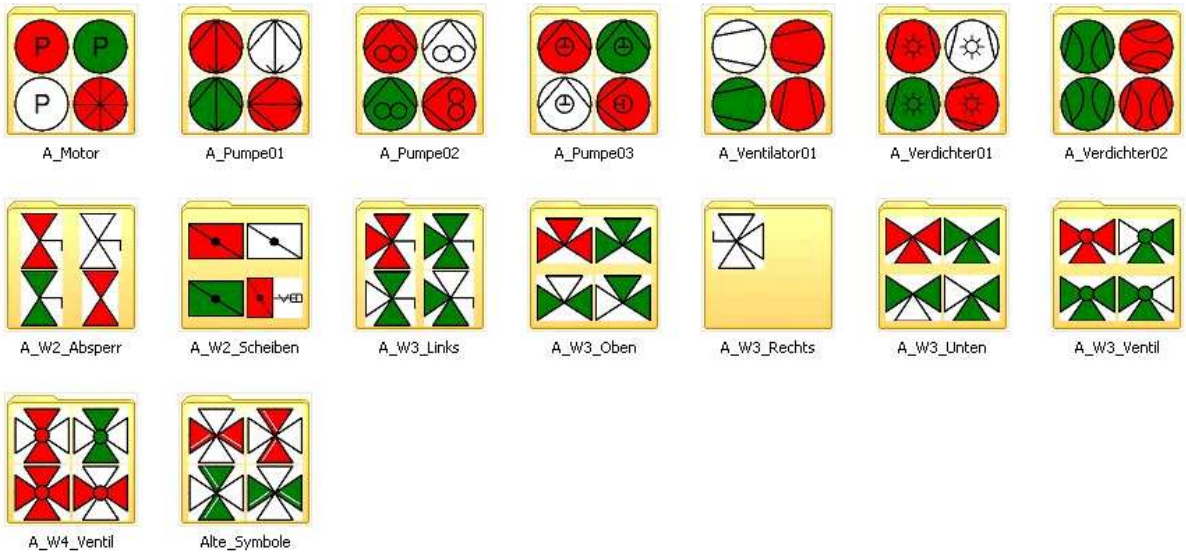
Die Bilddarstellung kann vom Anwender individuell verändert werden.

Die Hintergrundfarbe kann im Dialogfeld ausgewählt werden.





Im Standard ist eine umfangreiche Bilddatei an Bildsymbolen enthalten



Der Anwender kann eigene Symbole hinzufügen oder vorhandene in der Farbgebung ändern.

9.5 Ausgabefeld für WinCC

Im Sm@rtLib wird das Ausgabefeld dazu verwendet Analogwerte im Bild darzustellen.

Über einen Doppelklick in der Objektpalette/Controls auf das AGU_OS_Lib_1.FBSymbol wird das Control ins Bild gelegt. Ein Doppelklick auf das Control im Bild öffnet den Parametrierdialog.

Sie können jetzt die Strukturvariable über den Variablendialog in WinCC auswählen. Passend hierzu wählen Sie jetzt den Instanztypen aus (C_REGLER, DOSIERER etc.). Damit werden die entsprechenden Variablenverbindungen am Control erzeugt. Zusätzlich wird ein Ereignis-Script generiert, welches den Verbindungsstatus der in VarState als String hinterlegten Variablen prüft.
Liegt ein ungültiger Verbindungsstatus vor, wird das Control schraffiert dargestellt.

Der Parameter "Sammelrelevant" bestimmt ob die Variablenverbindung von CollectValue in die Sammelanzeige (Störgrundführung) eingeht.

In der folgenden Tabelle sind alle Eigenschaften außer den Standardeigenschaften des Objektes aufgelistet:

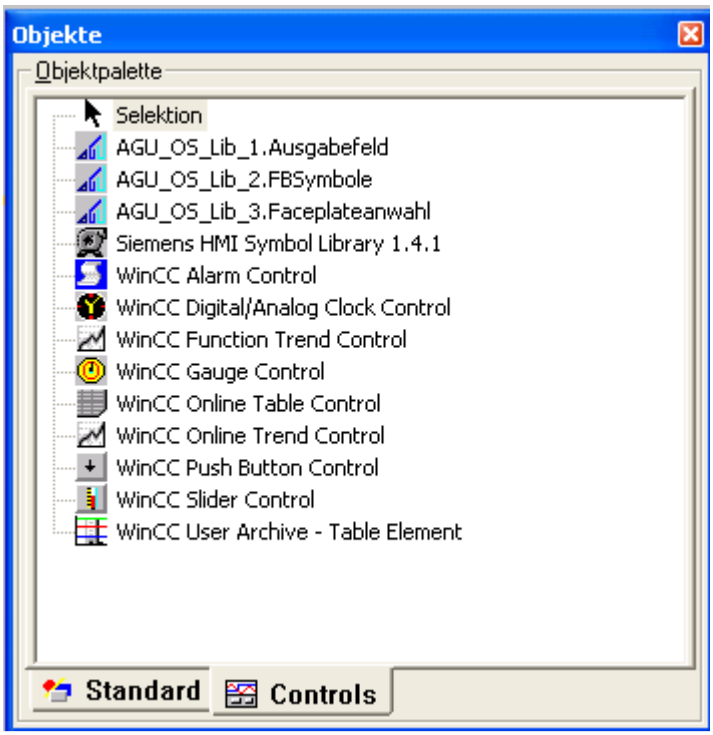
Wert	Wertbeschreibung	Eigenschafts-seite
Ausgabewert	Zeigt den Analogwert der Instanz an.	-
Relevant	Bestimmt ob die Variablenverbindung von CollectValue in die Sammelanzeige (Störgrundführung) eingeht.	1
CollectValue	Ereignisvariable der Instanz.	-
font	Einstellen der Schriftart des Ausgabefeldes.	-
Ausgabeformat	Bestimmt die Art der Anzeige des Wertes während der Runtime.	-
BackColor	Hintergrundfarbe	-
ForeColor	Schriftfarbe	-
AutoSize	Wenn Autosize aktiviert wird berechnet das Control automatisch seine Größe an Hand der in Ausgabeformat festgelegten Stellenanzahl.	
ZUST	Zustandsvariable der Instanz	
VarState	Die hier als String hinterlegte Variable wird im Ereignis-Script "VarStateAusgabewert" auf ihren Verbindugsstatus überprüft. Ist der Verbindungsstatus ungültig wird das Control schraffiert dargestellt. Das Ereignis-Script wird durch die Eigenschaftsmaske des Controls generiert.	
state	Steuert die Darstellung des Controls bei einem ungültigen Verbindungsstatus.	

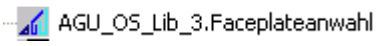
10 WinCC Grundlagen – ab SmartLib V4.1.0

Ab der Version 4.1 können Fließbildsymbole als „UDO“ genutzt werden. Dies ermöglicht es dem Anwender, seinen eigenen Zustandsfarben und Darstellungsformen im Fliesbild zu wählen.

10.1 Aufruftaste für das Faceplate

Auswahl des Controls zur Bildanwahl des Faceplates

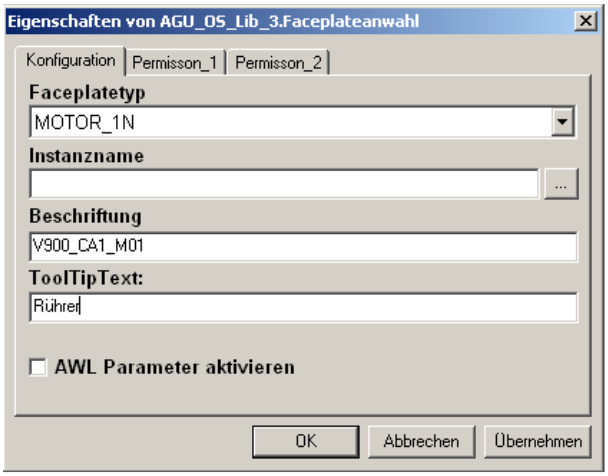


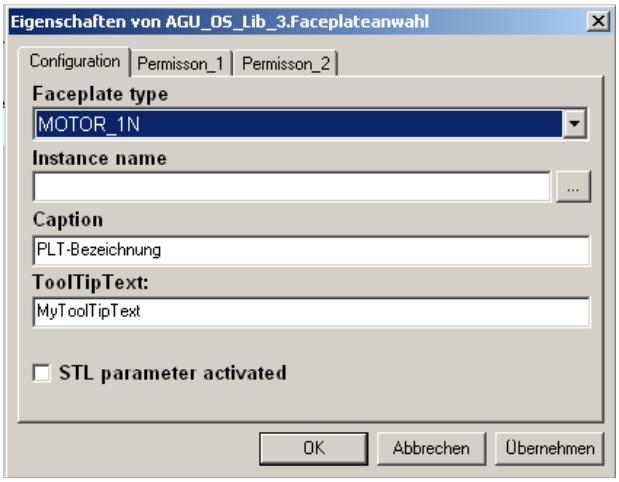
Mit dem Control 

kann der Anwender einen Faceplateaufruf realisieren. Das Control wird in das Anlagenbild gezogen und es erscheint die Standardtaste

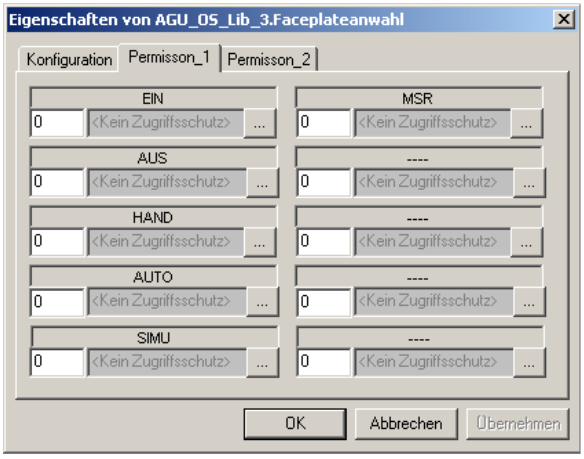
.

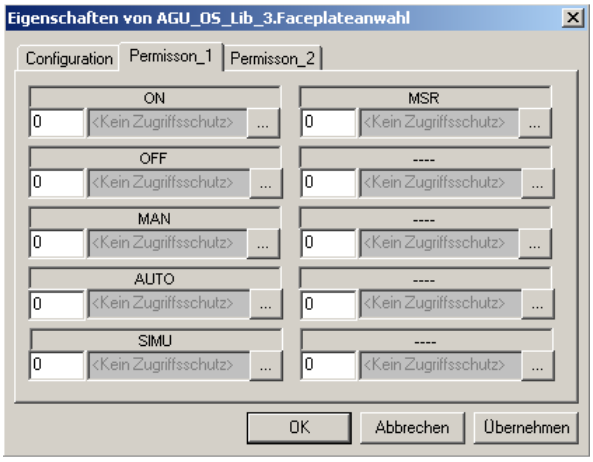
Durch einen Doppelklick auf die Standardtaste wird der Konfigurationsdialog aufgerufen:





Hier kann nun der Typ des Faceplates gewählt werden und der Instanzname aus dem Variablenhaushalt verschaltet werden.





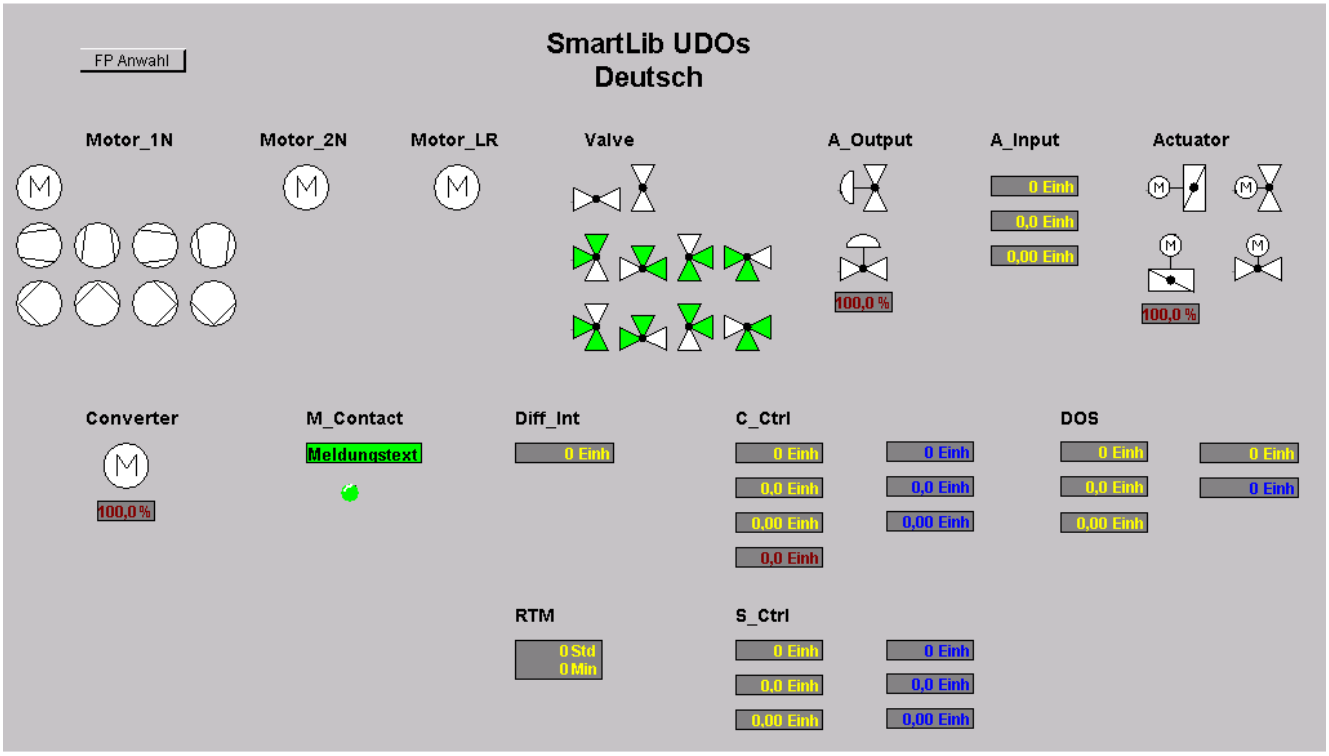
Der Anwender kann auf jede Funktion eine individuelle Bedienberechtigungsstufe legen.

11 WinCC Bildsymbole als UserDefinedObjects (UDO)

Die Symbole können projektspezifisch erstellt und angepasst werden. Dies ist in der Regel ohne VB Active-X Kenntnisse möglich. Die UDO Symbole können durch ein Tool ausgetauscht werden und sind in ihrer Gestaltung jederzeit änderbar.

11.1 Bildsymbole verwenden

Das Standardprojekt beinhaltet das Bild @AGU_SmartLibUDO_FB_Symbole.pdl. In diesem Bild finden Sie Beispiele der Bildsymbole als UDO. Die Symbole können zur Verwendung in den Anlagenbildern aus diesem Bild kopiert werden und mit dem Sm@rtLib-Konfigurator (ALT+S) dynamisiert werden.
Wenn Sie die Symbole auf Ihr Projekt anpassen wollen, dann benennen Sie das Bild um! Andernfalls könnte das Bild durch den Konfigurator überschrieben oder gelöscht werden!



Für eine projektspezifische Anpassung können die Symbole entsprechend der folgenden Beschreibung geändert werden.

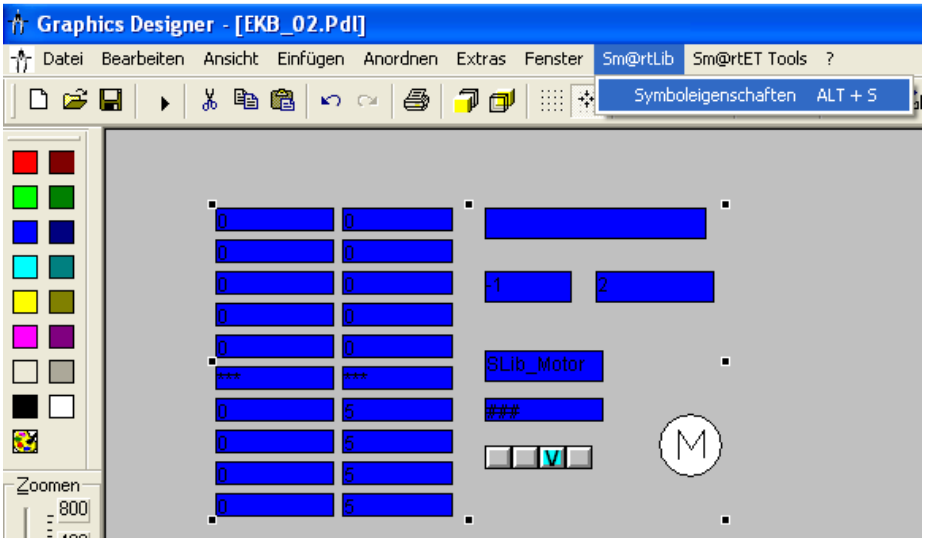
11.2 Grundlagen zum UDO

11.2.1 UDO: Beispiel - Motorbaustein

Am Beispiel des Motorbausteins wird der Aufbau des UDO erläutert. Im Bearbeitungsmodus können die unterschiedlichen Elemente des UDO dargestellt werden. Diese werden zu einem späteren Zeitpunkt klein gemacht und hinter der Zustandanzeige versteckt.

Darstellung UDO des Motors

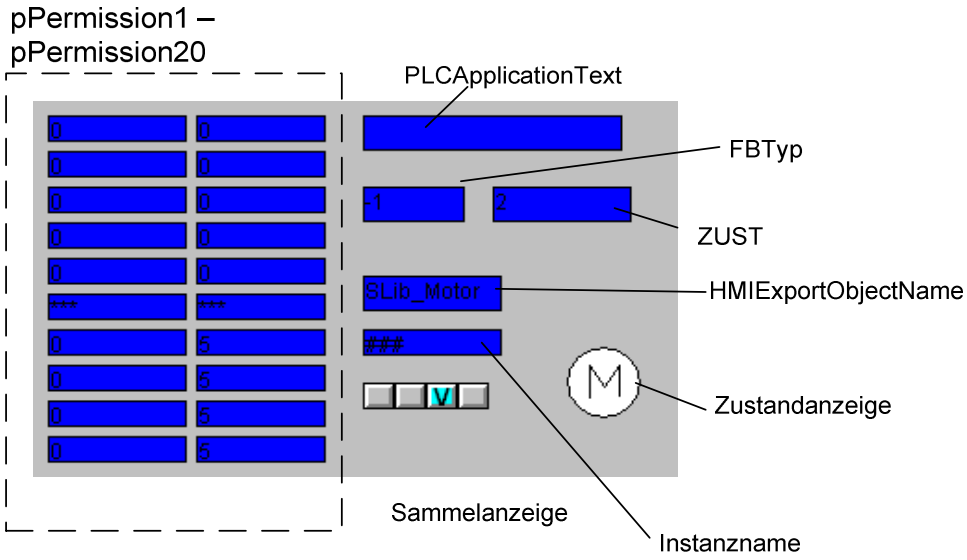
Zerlegt man das UDO und vergrößert die hinterlegten Felder, dann sieht man die im Bild gekennzeichneten Felder. Diese blauen Felder stellen die Schnittstelle zwischen der Sm@rtLib Standard Konfiguration (Anbindung der Variablennamens, Bedienberechtigungen, u.a.), dem Symbol und dem Faceplate dar.



Beispiel: Die im Konfigurationsdialog eingegebene Bedienberechtigung wird im verdeckten EA-Feld des UDO abgelegt. Im Runtime Modus klickt der Bediener auf das Symbol und öffnet darüber das Faceplate. In diesem Moment, wird die parametrierte Bedienberechtigung an das Faceplate übergeben.

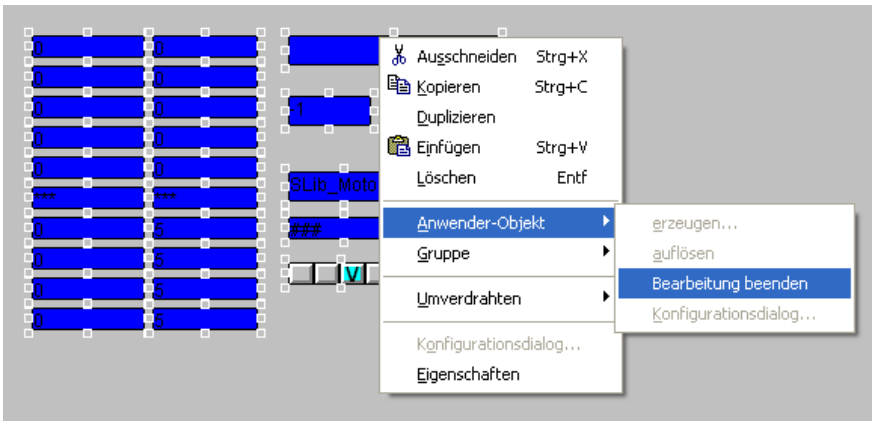
11.2.2 UDO: Motorbaustein – Aufbau des UDO

Die blauen Felder sind EA Felder, welche als Übergabeablage zum Konfigurationsdialog des UDO dienen. Die folgende Abbildung zeigt die Namen der Felder.



Jedes Feld hat einen Namen und die Eigenschaften des Feldes werden im Konfigurationsdialog des UDO später als Übergabeschnittstelle genutzt.

Wenn der Bearbeitungsmodus beendet wird, ist das UDO ein geschlossenes Element.

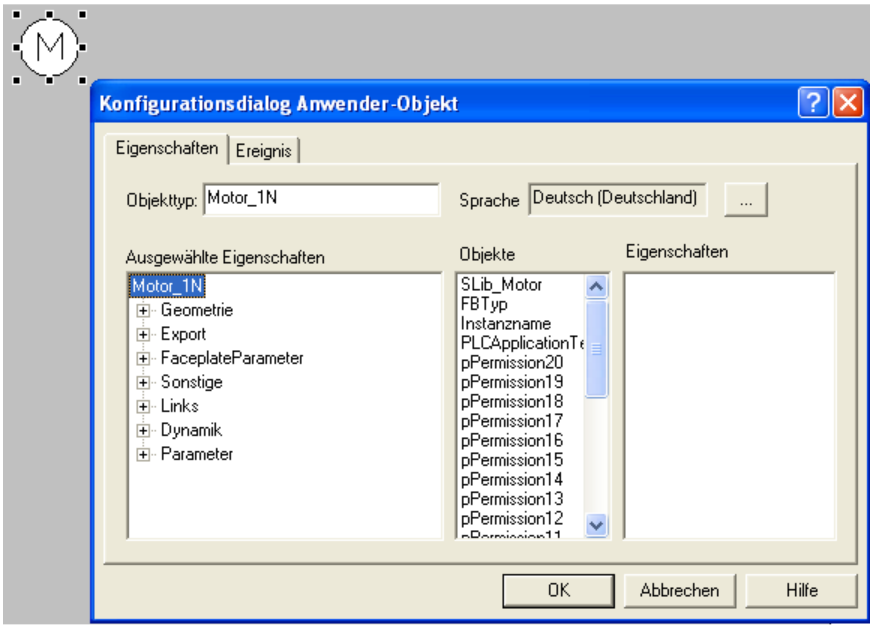


Neben den Feldern, ist die eigentliche Zustandsanzeige von wesentlicher Bedeutung. Der Anwender kann die Darstellung des Motorsymbols im Bild über diese Anzeige ändern.

11.2.3 UDO: Motorbaustein – interne Funktionen

Im Konfigurationsdialog des UDO werden die Eigenschaften des UDO sichtbar.

Das Feld „ausgewählte Eigenschaften“ zeigt alle Eigenschaften des UDO (Summe aller Teilelemente des UDO). Auf den einzelnen Eigenschaften liegen die Verbindungen zu den Elementen des UDO.

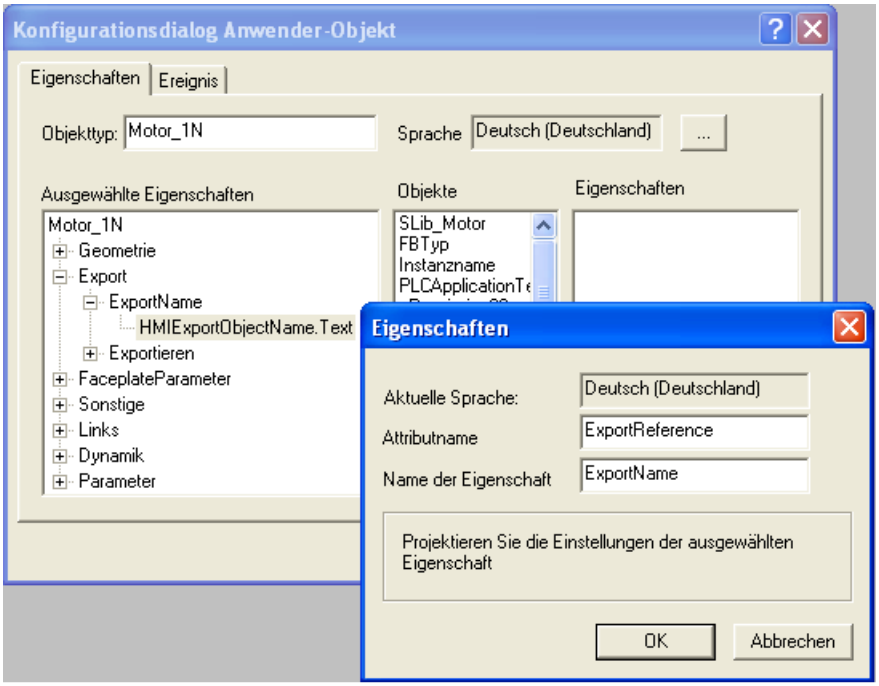


Diese Eigenschaften werden nun genutzt, um die Daten bei der Parametrierung zur Laufzeit an das Faceplate zu übergeben und die Zustandsanzeige über das Script zu versorgen.

Für jeden Typ des Sm@rtLib Standards gibt es ein spezielles Script, welches die Zustandanzeige mit dem Symbol verbindet.

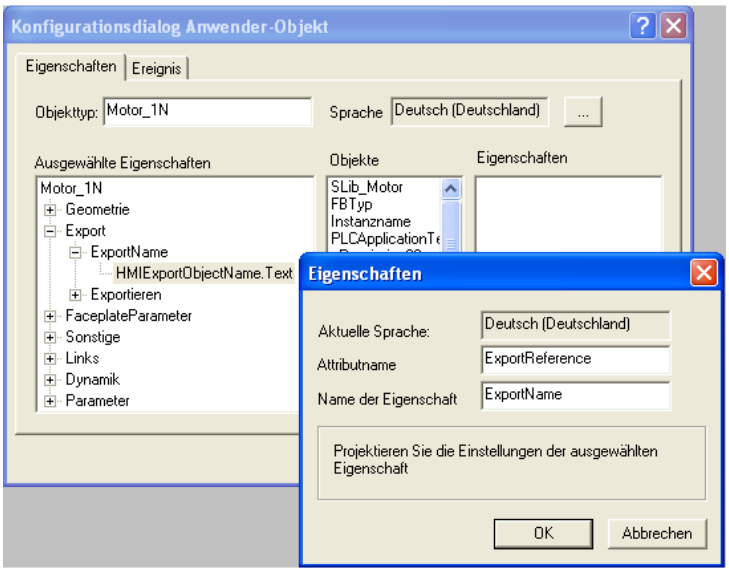
11.2.3.1 Standardteil der Funktion

Zum Beispiel liegt auf der Eigenschaft Export die Verbindung zu dem EA Feld „HMIExportObjectName“. Die Tabelle auf der folgenden Seite zeigt alle Eigenschaften und Namenszuordnungen!



Der Eigenschaftsdialog dieses Objektes zeigt einen Attributnamen und einen Namen der Eigenschaft, der aus den VB Script heraus aufgerufen wird (Also NIEMALS ÄNDERN).

Die EA Felder des UDO werden im Konfigurationsdialog mit den Eigenschaften verbunden.

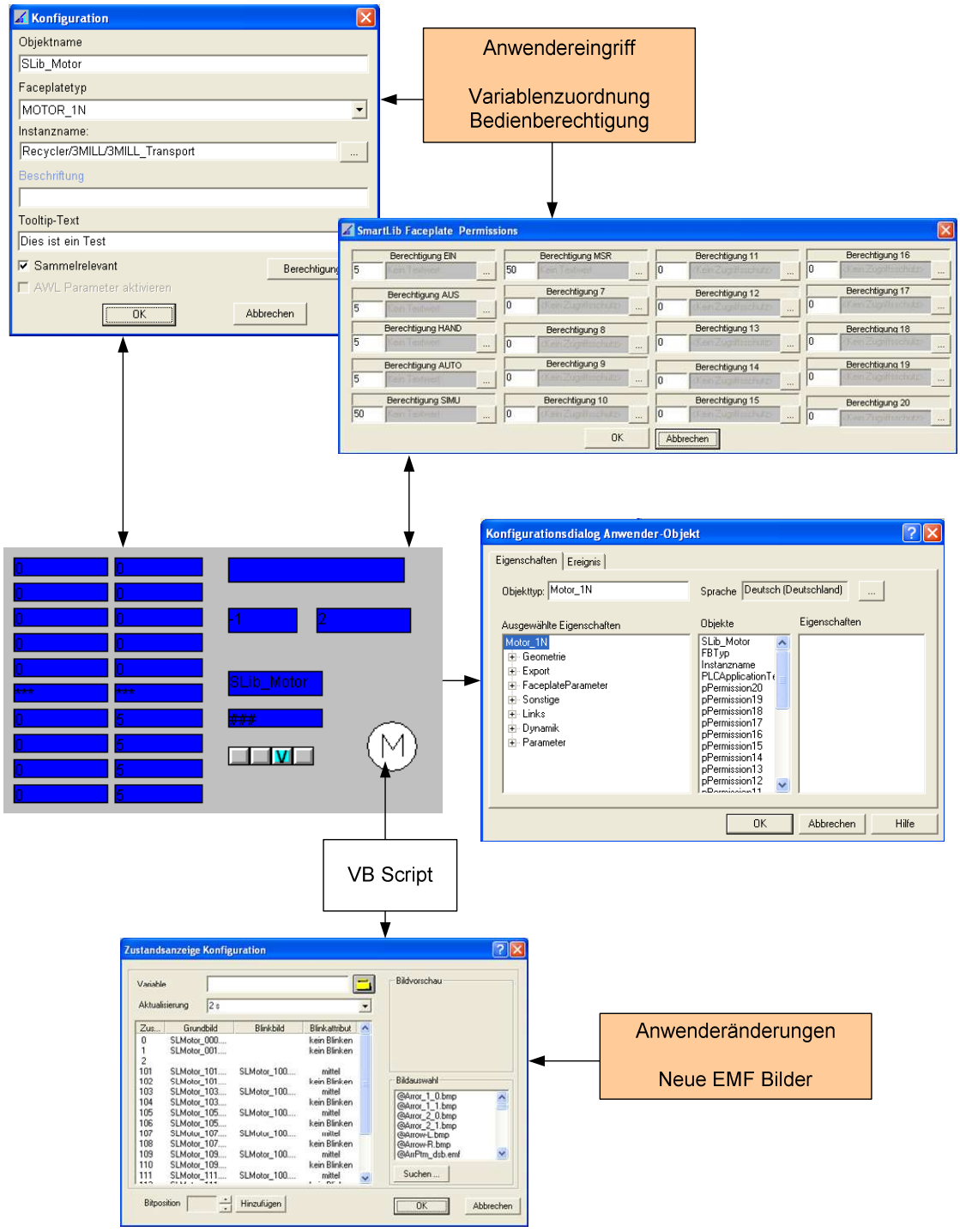


Spezifische Eigenschaften des UDO

Eigenschaft	Eigenschaftenname	EA Feldname im UDO	Attributname (!)	Name der Eigenschaft	
Export	ExportName	HMIExportObjectName.Text	ExportReference	Exportname	Symbolaustausch
	Exportieren	HMIExportObjectName.Blinken.Hintergrund aktiv	ExportEnable	Exportieren	Symbolaustausch
Faceplate-Parameter	Bedienberechtigung EIN	pPermission1.Ausgabewert	pPermission1	Berechtigung EIN	Berechtigung Faceplate
	…	…	…	…	…
	Bedienberechtigung …	pPermission20.Ausgabewert	pPermission20	Berechtigung 20	…
	PCLTextStream	PCLApplicationText.Text	TextStream	PCLTextStream	Faceplate Texte bei AWL Programmen
	Instanzname	Instanzname.Text	Instanzname	Instanzname	Übergabe des Instannamens an das Faceplate
	Faceplatetyp	FBTyp.Ausgabewert	Faceplatetyp	Faceplatetyp	Festlegung welche Faceplate geöffnet wird
	PCLApplication	PCLApplicationText.Fett	PCLApplication	PCLApplication	Das STEP7 Programm wurde in AWL erstellt. PLCTextStream ist relevant
Sonstige	Anzeige	Zustandanzeige.Anzeige	Visible	Anzeige	Default
Links	Sammelwert	Sammelanzeige.Sammelwert	CollectValue	Sammelwert	Verschaltung des EventState zur Bildung einer Sammelnazeige
	ZUST	ZUST.Ausgabewert	ZUST	ZUST	Verschaltung des Q_STATE zur dynamisierung der Betriebsarten
Dynamik	SymbolZustand	Zustandsanzeige.Aktueller Zustand	Index	SymbolZustand	Wird vom Script beschrieben zur dynamisierung der Zustandsanzeige
Parameter	Sammelrelevant	Sammelanzeige.Sammelrelevant	Relevant	Sammelrelevant	Messstelle wird für die Sammelanzeige aufgesammelt.

11.2.3.2 Der Zusammenhang - Parametrierung, Anzeige & Faceplate-Aufruf

Das UDP ist die Schnittstelle zwischen Offline Parametrierung und Online Anzeige und Faceplate-Aufruf. Die folgende Abbildung zeigt die prinzipiellen Verbindungen. Für den Anwender ist jedoch nur ein kleiner Ausschnitt von Interesse. Hier können Änderungen vorgenommen werden.



Der Anwender kann die Symboldarstellung jederzeit ändern und seine eigene Symboldatei anlegen!

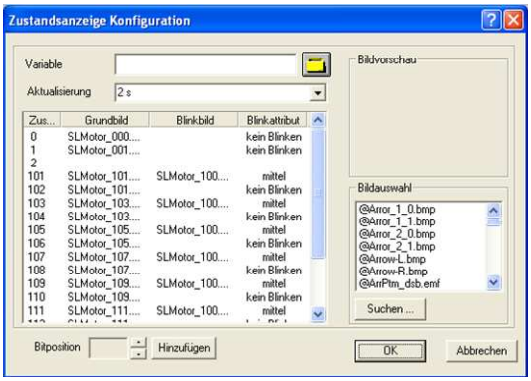
11.2.3.3 Darstellung im Bild ändern (Aktor)

Schritt 1: UDO kopieren in ein leeres Bild

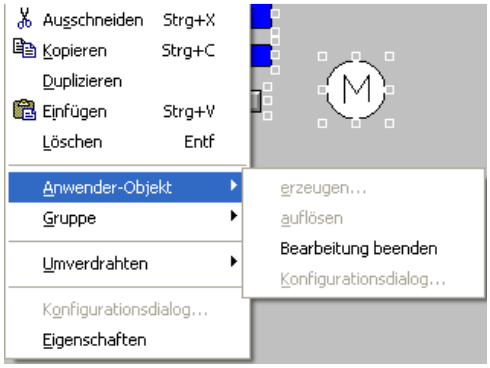
Schritt 2: Anwender-Objekt bearbeiten…

Schritt 3: Zustandanzeige anwählen und rechte Maus: Konfigurationsdialog

Schritt 4: Austauschen der EMF Dateien entsprechend der Farbgebung und Zahlenzuordnung



Schritt 5: Alle Objekte des UDO markieren und Anwender Objekt – Bearbeitung beenden



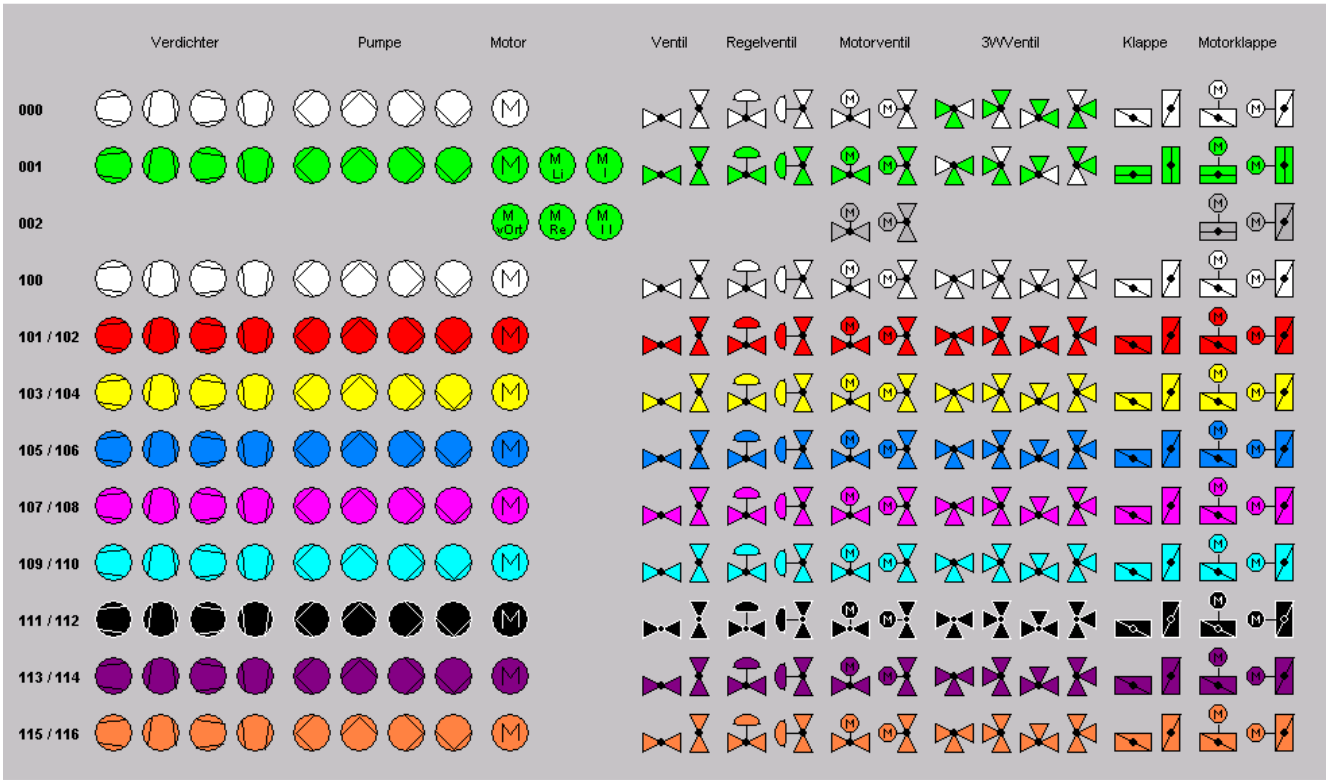
11.2.3.4 Farbdarstellung bei Aktoren

Bit	Meldeklasse	Meldeart	Farbe	SmartLib Index
31 (MS)	Alarm	AH	*	101
30	Alarm	AL	*	101
29	Warnung	WH	*	103
28	Warnung	WL	*	103
27	Toleranz	TH	*	105 (SLib n.v.)
26	Toleranz	TL	*	105 (SLib n.v.)
25	AS - Leittechnikmeldung	Störung	*	107
24	AS - Leittechnikmeldung	Fehler (VerrStoer)	*	109
23	Wartungsanforderung	Maintenance Request	*	111 (SLib n.v.)
22	Prozessmeldung	PM		---
21	nicht belegt	nicht belegt		---
20	Bedienanforderung	OR	*	113 (SLib n.v.)
19	Bedienmeldung	OM		---
18	Status-Meldung	Status AS		---
17	Status-Melding	Status OS		---
16	Messstelle gesperrt	X	*	114 (SLib n.v.)
15	Alarm	AH quittert		102
14	Alarm	AL quittiert		102
13	Warnung	WH quittiert		104
12	Warnung	WL quittiert		104
11	Toleranz n. quittierpflichtig	-		---
10	Toleranz n. quittierpflichtig	-		---
9	AS-Leittechnikmeldung	Störung quittiert		108
8	AS-Leittechnikmeldung	Fehler quittiert		110
7	Wartungsanforderung	MR quittiert		112 (SLib n.v.)
6	Prozessmeldung	PM quittiert		---
5	nicht belegt	nicht belegt		---
4	Bedienanforderung ist bei PCS 7 nicht quittierpflichtig	-		114 (SLib n.v.)
3	Bedienmeldung ist bei PCS 7 nicht quittierpflichtig	-		---
2	Status-Meldung ist bei PCS 7 nicht quittierpflichtig	-		---
1	Status-Meldung ist bei PCS 7 nicht quittierpflichtig	-		---
0 (LS)	Messstelle gesperrt ist bei PCS 7 n. quittierpflichtig	-		116 (SLib n.v.)

* = mit Blinkbild Status Grundstellung

Sm@rtLib beinhaltet ein Musterbild mit Vorlagen von Aktoren. Diese können vom Anwender kopiert und geändert werden.

Die neuen EMF Bilder müssen im WinCC Projektpfad von „GraCS“ liegen.

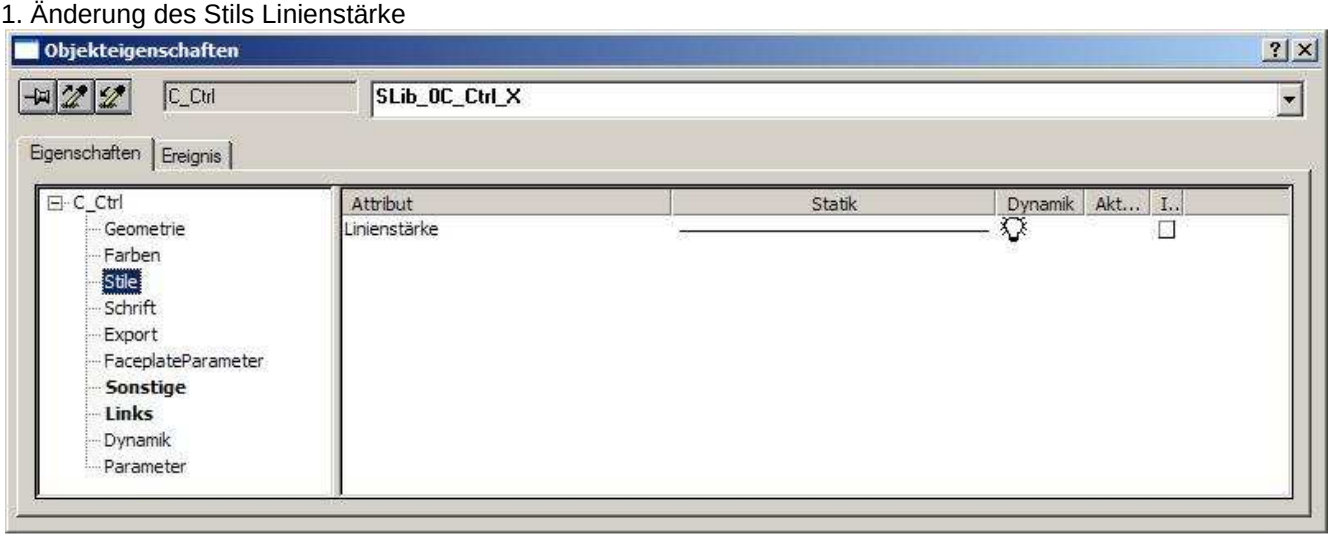


11.2.3.5 Darstellung im Bild ändern (Analogwert)

Die Darstellung eines Analogwertes im Bild kann größtenteils über die Eigenschaften des UDO geändert werden.

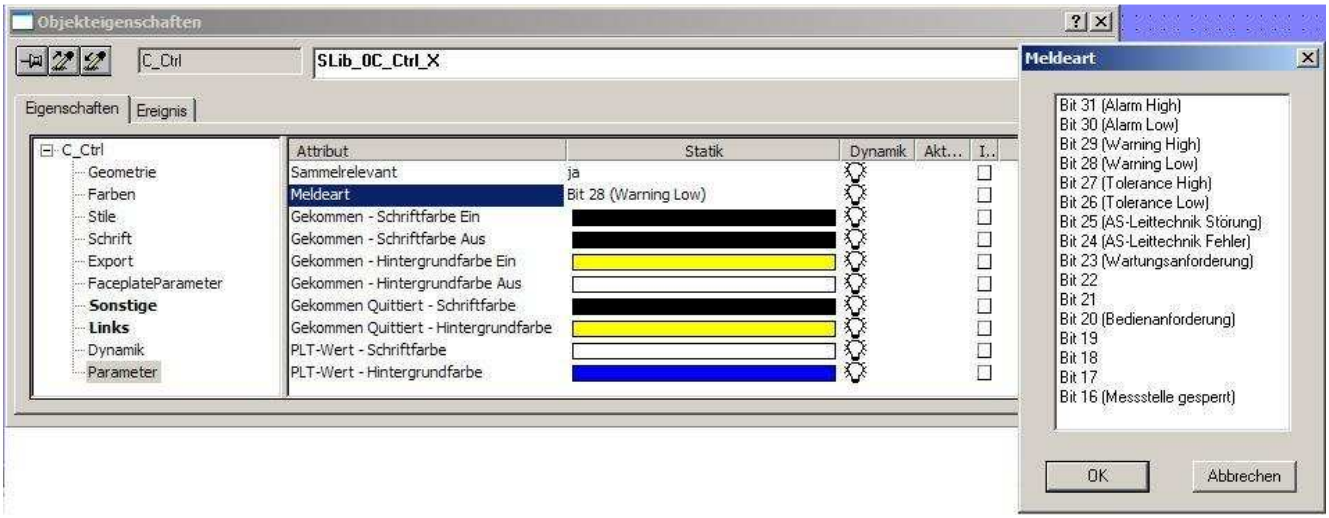
Zum Ändern muss das UDO angewählt werden und mit der rechten Maus die Eigenschaftsdialog geöffnet werden.

1. Änderung des Stils Linienstärke



2. Änderung der Schrift

3. Änderung der Farbgebung, bei den verschiedenen Meldearten.



Durch den Doppelklick auf die Zeile Meldeart öffnet sich die Auswahlbox der Meldeart. Nun kann die Meldeart ausgewählt werden und die gewünschte Farbgebung parametriert werden.

4. Änderung der Formatierung des Ausgabewertes (Anzahl der Stellen vor und hinter dem Komma)

Dazu muss das UDO bearbeitet werden

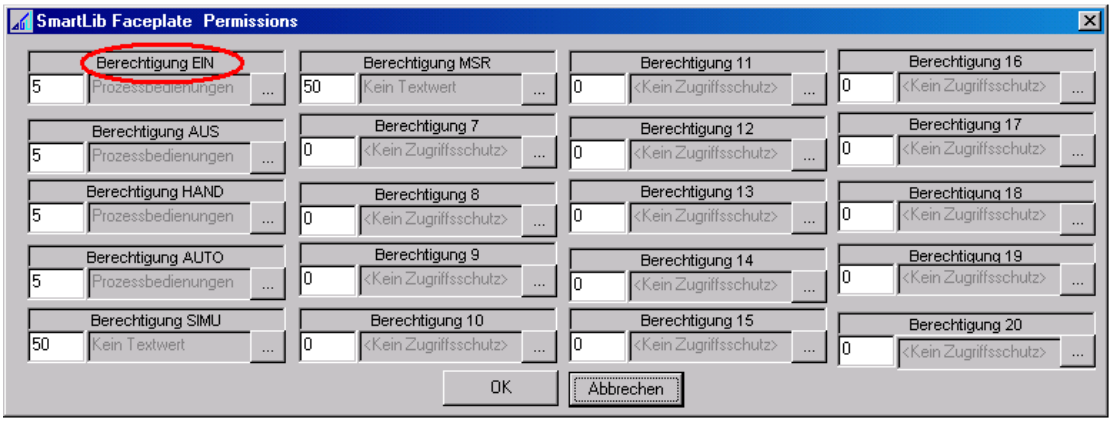
Nun das Element Wert markieren und unter den Eigenschaften das Ausgabeformat entsprechend parametrieren. In dem gezeigten Beispiel wird der Wert ohne Komma bis zu fünf Stellen dargestellt.

Eventuell muss bei einer anderen Formatierung des Ausgabewertes das Hintergrundfeld in seiner Größe angepasst werden. Das Hintergrundfeld muss mindestens so groß sein dass alle Stellen des Ausgabewertes hinterlegt sind!

Nach der Bearbeitung müssen wieder alle Elemente des UDO markiert werden und Bearbeitung beenden angewählt werden.

11.2.3.6 Anzeige der Bedienberechtigungsstufen konfigurieren

Damit im Konfigurationsdialog die möglichen Bausteinbedienungen als Text ausgegeben werden, muss für die Eigenschaft des UDO der Anzeigename für die Eigenschaft eingetragen werden,
Diese werden dann an den Konfigurationsdialog übergeben.



Wenn das Symbol auch in einem englischsprachigen Engineering-Projekt verwendet werden soll, dann müssen die Eigenschaftsnamen auch in Englisch vergeben werden.

11.2.3.7 Faceplate-Aufruf vom UDO

Das Faceplate wird über eine C-Aktion bei Mausklick auf das UDO ausgelöst. Dabei muss der Instanzname des Sm@rtLib-Bausteines an das Faceplate übergeben werden.

Für die Übergabe des Instanznamens an das Sm@rtLib-Faceplate gibt es zwei Möglichkeiten:

1.Auslesen der Eigenschaft Instanzname
Bei Mausklick wird die Funktion OpenFaceplate aufgerufen. Diese Funktion liest den Instanznamen aus der Eigenschaft „Instanz“. Die Eigenschaft Instanz ist ein Texteintrag, der bei mehrsprachigen Projekten für jede verwendete Sprache projektiert werden muss. Die Verknüpfung des UDO mit dem Konfigurator muss deshalb für alle benötigten Sprachen erfolgen.

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
{
#ifdef RUN_ON_WEBNAVIGATOR

//SetPictureName("WebScreen.pdl","Web_Kreisbild","@WEB_S7HVAC/RL01_M3/1.pdl");
//SetVisible("WebScreen.pdl","WEB_Kreisbild",TRUE);

#else

#pragma code("AGU_OS_Template.dll")
#include "AGU_OS_Template.h"
#pragma code()

OpenFaceplate

#endif


}
```

2.Auslesen des Instanznamens aus einer Variablen-Verbindung
Bei Mausklick wird die Funktion OpenFaceplateFromLink aufgerufen. Diese Funktion liest den Instanznamen aus dem, an einer definierten Eigenschaft verschalteten, Variablennamen. Dazu muss in der „#define LinkProperty“ Anweisung die Eigenschaft angegeben werden, an der diese Variable verschaltet ist.

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
{
#define LinkProperty "ZUST"

#ifdef RUN_ON_WEBNAVIGATOR

//SetPictureName("WebScreen.pdl","Web_Kreisbild","@WEB_S7HVAC/RL01_M3/1.pdl");
//SetVisible("WebScreen.pdl","WEB_Kreisbild",TRUE);

#else

#pragma code("AGU_OS_Template.dll")
#include "AGU_OS_Template.h"
#pragma code()

OpenFaceplateFromLink

#endif
}
```

Dies Variante ist bereits in den mitgelieferten Symbolen definiert (ab 28.3.2007).

12 Bedienmeldungen für WinCC

Bei jeder Bedienung eines SmartLib Bausteins über WinCC oder WinCCFlexible wird eine Bedienmeldung erzeugt. Die Bedienmeldung wird aus dem S7-Baustein gesendet wenn der Befehl im Baustein ausgeführt wurde. Die Texte in den Bedienmeldungen sind in den S7-Textbibliotheken OM_TXTLib und HVA_TXTLib enthalten.

12.1 Bedienmeldungen aus dem S7-Baustein

Für die S7-300 Version sind die Informationen in den Bedienmeldungen begrenzt.
Folgende Informationen werden übertragen:

	S7-300	S7-400
Name der Bausteininstanz	x	x
Bediente Funktion:	x	x
Neuwert / Altwert / Einheit :	-	x

12.1.1 OS-ID einrichten

Die erweiterten Bedienmeldungen sind nur für die S7-400 Version verfügbar, da die S7-300 Steuerungen über zu wenig Begleitwerte bei den Meldungen verfügen.
Das erweiterte Bedienprotokoll bietet die Möglichkeit, die WinCC-OS zu ermitteln von der aus bedient wurde.

Damit die OS-Kennung protokolliert werden kann, muss auf jeder WinCC-OS eine neue Variable eingefügt werden.

Legen Sie eine ínterne Variable mit dem Namen *@SLibOSID* an. Der Variablentyp ist ein *Vorzeichenloser 16-Bit Wert*. Stellen Sie die *Rechnerlokale Aktualisierung* ein.
Diese Variable muss zur WinCC-Runtime die OS-Kennung enthalten. Als OS-Kennung sind Zahlen zwischen 1 und 255 möglich. Als Kennung kann z.B. das letzte Octet der IP-Adresse verwendet werden.

Bei Serverprojekten die über einen Engineeringplatz geladen werden, kann z.B. in das Übersichtsbild @Overview.PDL bei Bildanwahl der Rechnername ausgewertet werden um die entsprechende OS-Kennung einzutragen.

12.2 Bedienmeldungen über WinCC (ab V4.1.1.3)

Seit der Version V4.1.1.3 werden zusätzlich zu den Bedienmeldungen aus dem S7-Baustein, auch Bedienmeldungen vom Faceplate erzeugt. Diese werden dann direkt in das WinCC Alarm-Logging geschrieben. SmartLib verwendet dafür die Standard WinCC-Meldung *1013001*. Diese wird vom OS-Projekteditor automatisch angelegt.

Folgende wesentliche Informationen enthält das Bedienprotokoll:

1. Benutzername
2. Altwert und Neuwert bei Wertveränderungen (Sollwert, Grenzwert,...)
3. Ausführungsquittierung bei Bedienungen (EIN, AUS, Hand, Auto, Halt,…)

12.2.1 Bedienung einer Funktion ohne Wertänderung:

In diesem Fall wird die Bedienung direkt vom Faceplate aus in das Bedienprotokoll eingetragen. Die Bedienung löst im Funktionsbaustein die Bedienfunktion aus. Der Funktionsbaustein in der S7 Steuerung sendet nach der Ausführung eine Bedienquittierung in das Bedienprotokoll.

Wurde eine Funktion bedient, ohne das die Bedienung ausgeführt wurde, wird dies zwar im Bedienprotokoll eingetragen, aber die Bedienquittierung aus der S7 fehlt. In dem Beispiel wurde die Taste EIN bedient aber nicht ausgeführt. Dagegen wurde HALT bedient und auch ausgeführt.
Die Quittierung QS wird immer nur als einfach in das Bedienprotokoll eingetragen.

12.2.2 Bedienung mit Wertänderung

Eine Bedienung mit Wertveränderung wird ebenfalls direkt in das Bedienprotokoll eingetragen. Die Bedienung wird mit Alt- und Neuwert protokolliert. Die Ausführung der Bedienung in der S7 wird durch eine Bedienquittierung im Bedienprotokoll eingetragen.

Einzelne Bausteine (z.B. SCHEDULER und CLOCKTIMER) können Besonderheiten in der Bedienprotokollierung aufweisen.

12.2.3 Texte in Bedienmeldungen

Die Texte in den Bedienmeldungen für die bediente Funktion entsprechen immer dem Text auf der bedienten Taste bzw des Eingabefeldes. Es wird immer der Text der aktuellen Runtimesprache archiviert. Bei einer späteren Sprachumschaltung in Runtime, bleibt der Text in der Sprache die zum Zeitpunkt der Bedienung eingestellt war.

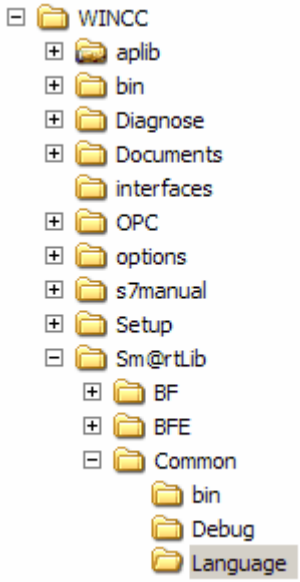
13 Sprachumschaltung projektweit

Sie können die Sprache projektweit einstellen. Dazu gehen Sie in den Step7 Manager und WinCC Projektmanager und stellen die Sprache in den Optionen ein.

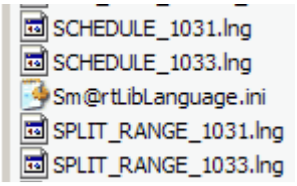
13.1 Neue Sprache für WinCC Faceplates ergänzen

Die Sprache kann durch den Anwender selber verändert werden. Dazu müssen Sie die neuen Textdateien in dem folgenden Pfad speichern.

Ordner: ...\Siemens\WinCC\Sm@rtLib\Common\Language



Sie finden in diesem Pfade die Textdatei der aktuell vorhandenen Sprachen und die zentrale Verwaltungsdatei Sm@rtLibLanguage.ini. Die Zahlenkennzeichnung entspricht der Windows Länderkennung. Kopieren Sie sich die entsprechenden Textdateien und benennen Sie diese um (neuer Zahlencode). Tragen Sie die neuen Textdateien nach dem Kopieren in die zentrale Verwaltungsdatei Sm@rtLibLanguage.ini ein.



zentrale Verwaltungsdatei Sm@rtLibLanguage

[LANGUAGE]
1=GERMAN,1031,
2=ENGLISH,1033,
3=FRENCH,1036,
4=SPANISH,1034,
5=ITALIAN,1040,

[GERMAN]
DOSIERER=DOSIERER_1031.lng
FW_CONTROL=FW_CONTROL_1031.lng
DIFF_INT=DIFF_INT_1031.lng
ANAAUS=ANAAUS_1031.lng
ANAEIN=ANAEIN_1031.lng

C_REGLER=C_REGLER_1031.lng
MOTOR_1N=MOTOR_1N_1031.lng
UMRICHTER=UMRICHTER_1031.lng
Ventil_AZ=Ventil_AZ_1031.lng
MASTER=MASTER_1031.lng
SPLIT_RANGE=SPLIT_RANGE_1031.lng
ZAMKlappe=ZAMKlappe_1031.lng
SCHEDULE=SCHEDULE_1031.lng
RUN_TIME_METER=RUN_TIME_METER_1031.lng
MELDUNG=MELDUNG_1031.lng
KonfigEditor=KonfigEditor_1031.lng
OS_Lib_Symbols=OS_Lib_Symbols_1031.lng

[ENGLISH]
DOSIERER=DOSIERER_1033.lng
FW_CONTROL=FW_CONTROL_1033.lng
DIFF_INT=DIFF_INT_1033.lng
ANAAUS=ANAAUS_1033.lng
ANAEIN=ANAEIN_1033.lng
C_REGLER=C_REGLER_1033.lng
MOTOR_1N=MOTOR_1N_1033.lng
UMRICHTER=UMRICHTER_1033.lng
Ventil_AZ=Ventil_AZ_1033.lng
MASTER=MASTER_1033.lng
SPLIT_RANGE=SPLIT_RANGE_1033.lng
ZAMKlappe=ZAMKlappe_1033.lng
SCHEDULE=SCHEDULE_1033.lng
RUN_TIME_METER=RUN_TIME_METER_1033.lng
MELDUNG=MELDUNG_1033.lng
KonfigEditor=KonfigEditor_1033.lng
OS_Lib_Symbols=OS_Lib_Symbols_1033.lng

14 Automatisches Erzeugen von Bildsymbolen im PCS7

Bei PCS7 kann das automatische Erzeugen der Bildsymbole beim AS-OS-Transfer auch mit SmartLib genutzt werden.

Während der Installation von SmartLib wird das Bild *@PCS7Typicals_AGU_SmartLibUDO_FB_Symbole.pdl* installiert. Das Bild wird beim Einrichten des WinCC-Projektes automatisch in das GraCs-Verzeichnis von WinCC kopiert. Der Bildname des Typicalbildes muss immer mit *@PCS7Typicals_* beginnen, damit das Bild beim AS-OS-Transfer gefunden wird.

Die Vorgehensweise beim automatischen Erzeugen der Bildsymbole ist im PCS7-Engineeringhandbuch beschrieben. Im CFC-Editor muss ein Symbolname für die Bausteininstanz eingegeben werden.
Der Name eines Bildsymbols findet sich im Eigenschaftsdialog des Symbols unter der Eigenschaft **Export/type*.***

Beispiel: Bausteinname/Symbolname
@C_CTRL_41/SLib_0C_Ctrl

Der Bausteinname ergibt sich aus dem Namen des SmartLib-FB und muss nicht mit eingetragen werden.

15 Darstellungen und Bedienung

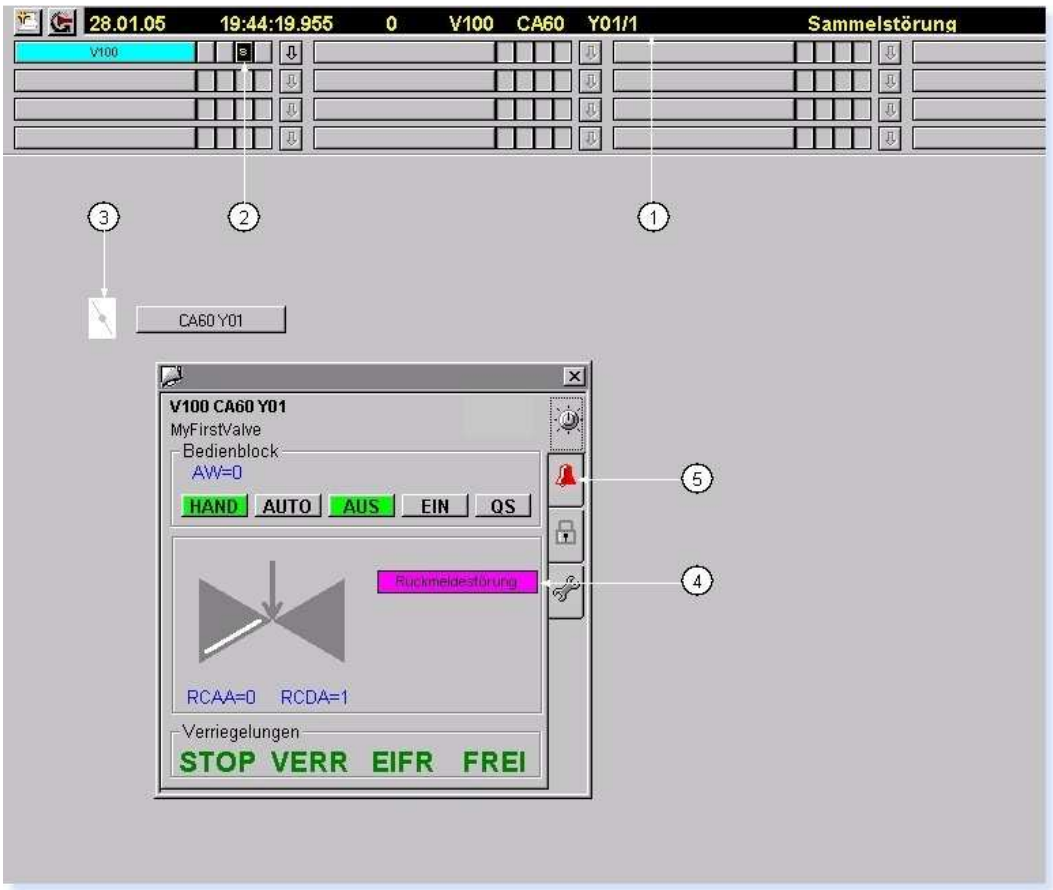
In diesem Abschnitt wird die Bedienung der Faceplates am Beispiel ausgewählter Bausteine erläutert. Die aufgeführten Funktionen sind bei allen Bausteinen und Bedienelementen gleich und wiederholen sich.

15.1 Störanzeige in WinCC

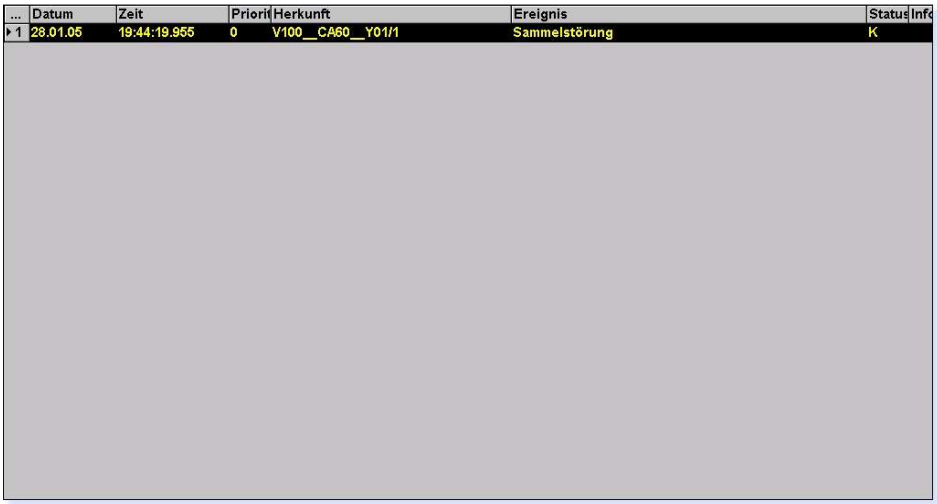
Im Falle einer Störung, wird diese an unterschiedlichen Stellen visualisiert.

Die Störung wird an folgenden Stellen angezeigt:

1. Meldezeile im Übersichtsbereich des Alarm-Logging
2. Störgrundführung ins Fliessbild über den Übersichtsbereich
3. Störungssicht im Fliessbild durch das Fliessbildsymbol
4. Detailansicht der Störung im Faceplate
5. Störungsanzeige der Instanz im Alarm-Logging

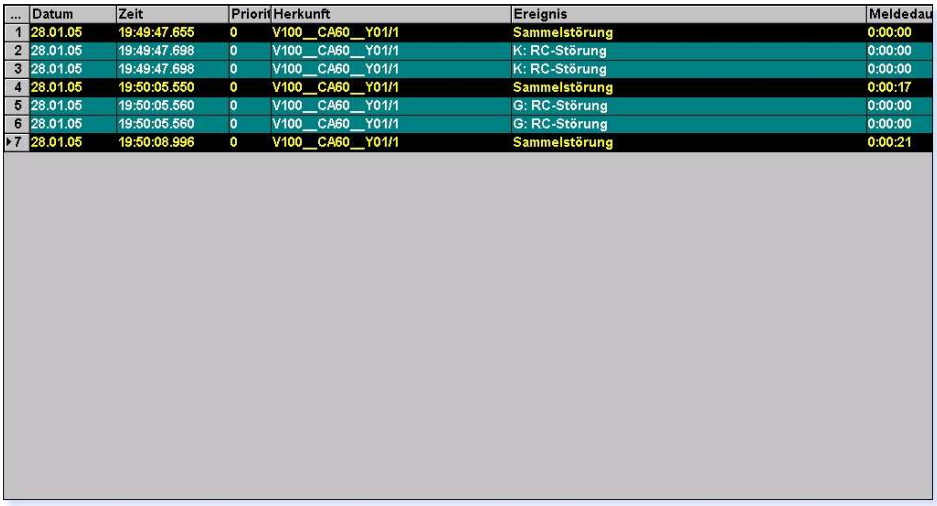


Im Alarm-Logging wird die Störung mit dem Zeitstempel der Steuerung in der "Neuliste" angezeigt.



...	Datum	Zeit	Priorit	Herkunft	Ereignis	Status	Info
1	28.01.05	19:44:19.955	0	V100__CA60__Y01/1	Sammelstörung	K	

Der Meldeverlauf wird im Alarm-Logging in der "Chronikliste" wie folgt dargestellt.



...	Datum	Zeit	Priorit	Herkunft	Ereignis	Meldedau
1	28.01.05	19:49:47.655	0	V100__CA60__Y01/1	Sammelstörung	0:00:00
2	28.01.05	19:49:47.698	0	V100__CA60__Y01/1	K: RC-Störung	0:00:00
3	28.01.05	19:49:47.698	0	V100__CA60__Y01/1	K: RC-Störung	0:00:00
4	28.01.05	19:50:05.550	0	V100__CA60__Y01/1	Sammelstörung	0:00:17
5	28.01.05	19:50:05.560	0	V100__CA60__Y01/1	G: RC-Störung	0:00:00
6	28.01.05	19:50:05.560	0	V100__CA60__Y01/1	G: RC-Störung	0:00:00
7	28.01.05	19:50:08.996	0	V100__CA60__Y01/1	Sammelstörung	0:00:21

Zuerst kommt die Sammelstörung ohne Störgrundkennung. Der Störgrund wird als Betriebsmeldung mit Kennung "K := kommend" nachgeliefert. Dann wird die Sammelstörung durch den Bediener im Alarm-Logging quittiert. Das Rücksetzen der Störung über den QS-Button im Faceplate beendet den Störzustand. Über die Betriebsmeldung mit der Kennung "G := gehend" wird die Störung als gehend protokolliert. Als letztes geht die Sammelstörung.

15.2 Allgemeine Anzeigen im Faceplate

Ein Faceplate dient zur Bedienung von Anlagenfunktionen, die einen großen Wiederholungsgrad haben.

Die Faceplates haben generell einen gleichen Grundaufbau und sind daher von der Funktionsweise und den gegebenen Informationen für den Anwender leicht überschaubar. Dieser gleiche Aufbau ermöglicht es dem Bediener eine einmal erlernte Bedien- und Informationsstruktur eines Bedienelementes auf ein anderes Bedienelement zu übertragen.

Die Anwahl des Faceplates geschieht über den entsprechenden Anwahlbutton oder direkt auf dem Symbol.

Beispiel einer typischen Faceplate-Anwahl:

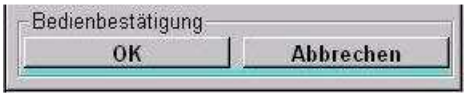
Ventil, Anwahl über Anwahlbutton oder Bildsymbol

15.2.1 Grundaufbau

- Kopfzeile
  In der Kopfzeile stehen Messstellenbezeichnung (AKZ) und Messstellennname
- Bedienblock
  Im Bedienblock werden die Bedienungen ausgeführt und der aktuelle Betriebszustand angezeigt.
- Status.- und Zustandsanzeige
  In der Status.- und Zustandsanzeige werden die Rückmeldungen und anstehende Meldungen angezeigt.
- Verriegelungen
  In dem Verriegelungsfenster werden die Zustände der Verriegelungen angezeigt.
- Bedienungsbestätigung (nach erfolgter Bedienung)
  Die Bedienbestätigung wird nach einer Bedienung, im Verriegelungsfenster eingeblendet. Hier kann die Bedienung bestätigt oder abgebrochen werden.

Der Grundaufbau eines Faceplates wird am Beispiel des Faceplates des Ventils dargestellt:

In der Abbildung unten wird die Bedienungsbestätigung exemplarisch gezeigt.



Dieses Feld ist für alle Faceplates gleich und wird in das Verriegelungsfeld eingeblendet, wenn eine Bedienung über den Bedienblock erfolgt.

Hier muss zum Ausführen einer Bedienung, diese bestätigt werden. Ein Abbrechen der Bedienung ist ebenfalls möglich.

Im Folgenden wird dieses Feld nicht mehr eingeblendet, da die Funktionsweise dieses Feldes bereits bekannt ist.

15.2.2 Bedienblock und Betriebsartenteil

Jeder Aktor hat die Betriebsart Hand und Automatik. Durch einen grünen Farbumschlag wird der aktuelle Betriebszustand in den Buttons **HAND** bzw. **AUTO** angezeigt.

15.2.2.1 Handbetrieb

In dem oben gezeigtem Beispiel ist die Betriebsart **Hand** aktiv.

Über die Tasten **EIN** bzw. **AUS** kann der Aktor im Handbetrieb Ein und Aus geschaltet werden.

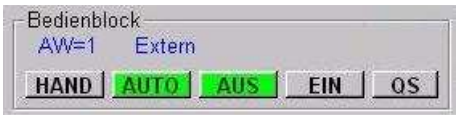
In die Betriebsart Hand kann nur gewechselt werden, wenn der Aktor nicht auf **Extern** steht.

15.2.2.2 Automatikbetrieb

Im Automatikbetrieb wird der Aktor von einer übergeordneten Instanz z.B. Schrittkette oder Regler geschaltet.

Die für den Automatikbetrieb relevanten Zustände, **Extern** bzw. **AW** werden als blauer Text über den Bedientasten eingeblendet.

Der Text „AW = 1“ bzw. AW = 0“ zeigt den Zustand des Automatikwertes an. Bei AW = 1 soll der Aktor über den Automatikbetrieb einschaltet werden. Bei AW = 0 soll der Aktor ausgeschaltet werden.



In dem oben gezeigten Beispiel steht der Aktor auf **Auto** und **Extern**, das heißt er kann nicht über das Faceplate eingeschaltet werden. Der **AW** steht auf 1, das heißt eigentlich müsste der Aktor eingeschaltet sein. Laut Rückmeldung auf der Taste **AUS** ist er aber aus. Dieser Betriebszustand lässt darauf schließen, dass der Aktor eine Störung hat (siehe Karteireiter Störung).

Für Testzecke können die Sm@rtLib-Bausteine auf die Betriebsart Simulation geschaltet werden. Bei aktiver Simulation werden keine Rückmeldestörungen oder Analogwertfehler gemeldet. Die Simulation wird im Bedienblock oben links mit dem Eintrag

angezeigt.

Im normalen Betriebszustand sollte diese Anzeige nicht zu sehen sein!

15.2.3 Karteireiter

Am rechten Rand des Faceplates sind diverse Karteireiter angebracht.
Die Karteireiter variieren je nach Bedienelement (Faceplatetyp).
In der Regel sind jedoch meistens vier Karteireiter vorhanden. Diese werden im Folgenden beschrieben. Spezielle Karteireiter werden bei der Erläuterung der entsprechenden Faceplates beschrieben.

- **Bausteinstatus**
- **Alarm**
- **Verriegelungen**
- **Service**

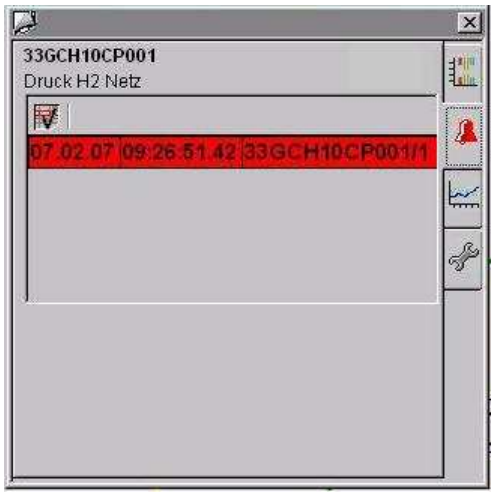
15.2.3.1 Bausteinstatus

Im Bausteinstatus wird, die Sicht auf den Statusteil des Faceplates ausgewählt.
Hier werden die vom Baustein verarbeiteten Rückmeldungen und anstehende Meldungen angezeigt.
Im oben gezeigten Beispiel sind folgende Rückmeldungen dargestellt:

RCAA = „Rückmeldung aktive Ansteuerung“ mit der Wertigkeit „0“
RCDA = „Rückmeldung deaktive Ansteuerung“ mit der Wertigkeit „1“

Je nach Faceplatetyp werden verschiedene Meldungen mit dem entsprechenden Farbumschlag der Meldeklasse angezeigt:

15.2.3.2 Alarm



Hier wird die Sicht auf den Meldestatus des Bausteins ausgewählt.
Das Glockensymbol blinkt im Falle einer nicht quittierten Meldung oder ist rot, wenn eine Meldung quittiert ansteht.

15.2.3.3 Verriegelungen

Hier werden die Verriegelungen als Texte im Detail dargestellt.

Der Gutzustand wird immer als Grüner Text und der Schlechtzustand als Weißer Text dargestellt.

STOP:
Der Baustein wird sofort ausgeschaltet, wenn der STOP-Befehl kommt. Wurde der Baustein durch den STOP-Befehl ausgeschaltet, so muss er wieder durch den Operator bzw. durch eine überlagerte Steuerung eingeschaltet werden. Das Einschalten kann aber erst dann erfolgen, wenn der STOP-Befehl nicht mehr am Baustein ansteht. Ein STOP-Befehl ist nicht quittierungspflichtig. Das heißt bei Auslösung des Stop-Befehls wird auch keine Meldung generiert.

VERR:
Sofern der Baustein eingeschaltet ist und ein Verriegelungs-Befehl kommt, wird der Baustein ausgeschaltet und geht in den Status *Verriegelungsstörung*. Eine solche Störung muss durch den Operator mit dem Button "QS" im Faceplate quittiert werden. Erst nach quittiertem und nicht mehr anstehendem Verriegelungs-Befehl kann das Element wieder eingeschaltet werden.

EIFR:
Wenn der Baustein eingeschaltet werden soll, müssen alle an ihm parametrierten Einschaltfreigaben im Gut-Zustand sein, ansonsten kann der Baustein nicht eingeschaltet werden. Geht eine Einschaltfreigabe, wenn das Element schon eingeschaltet ist, so hat dies keine Auswirkung auf den Zustand des Elements.

FREI:

Wenn der Baustein eingeschaltet wird und eine parametrierte Freigabe (ENAB) ist nicht erfüllt, so geht das Element intern erst dann in den eingeschalten Zustand, wenn die Freigabe erfüllt ist. Ein erneutes Einschalten des Elements ist dabei nicht notwendig.

15.2.3.4 Service

Hier wird die Sicht der Servicedarstellung angewählt. Sie ist für den allgemeinen Bediener nicht relevant.

Neben der Taste **Simulation** kann der Versionsstand der installierten Sm@rtLib Version abgelesen werden.

Ebenso kann unter **Blocktyp** der eingesetzte Bausteintyp abgelesen werden.

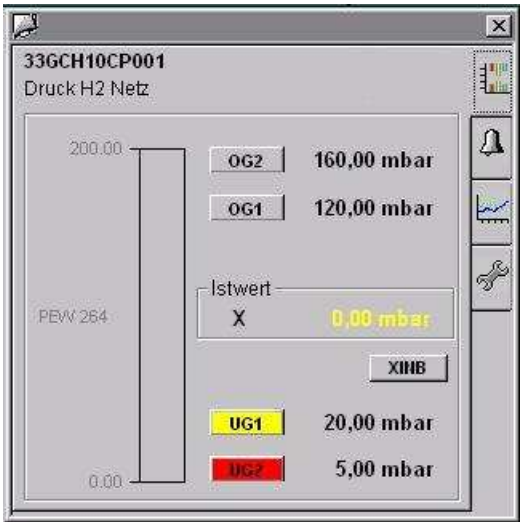
15.2.4 Werteingabe

Wird in den Faceplates ein Werteingabe Button angeklickt, erscheit eine Werteingabebox neben dem Faceplate. Hier kann der gewünschte Wert direkt über die Tastatur oder über die Pfeiltasten eingegeben werden.
In der Werteingabebox wird der einzugebende Wert (Y oder OG1) mit entsprechender Einheit sowie dem Wertbereich angezeigt.

Beispielhaft sind hier die Werteingabeboxen für die Stellgrößenvorgabe der Analogausgabe, und der Warngrenze Oben OG1 der Analogeingabe aufgeführt.

15.3 Analogeingabe

Der Baustein dient zum Einlesen, Beobachten und Überwachen eines Messwertes (Analogwertes). Es können vier Grenzwerte überwacht und parametriert werden. Sie können schaltend und oder alarmierend wirken. Störungen des Eingangssignals werden gemeldet. Der eingelesene Wert wird auf die Bereichsgrenzen skaliert.



Im Statusfenster des Faceplates wird der Istwert des Messwertes in Balkenform, mit der im Baustein parametrierten Skalierung, dargestellt. Ebenso wird der Istwert als Zahlenwert dargestellt.
Am Balken Anfang und Ende stehen jeweils der parametrierte Messbereichsanfang und das Messbereichsende.

Es können bis zu 2 oberer Grenzwerte definiert werden. Diese Grenzwerte können über die Button **"OG2"** bzw. **"OG1**" im Faceplate verändert werden. Die Grenzwerte können nur mit der entsprechenden Bedienberechtigung verändert werden!

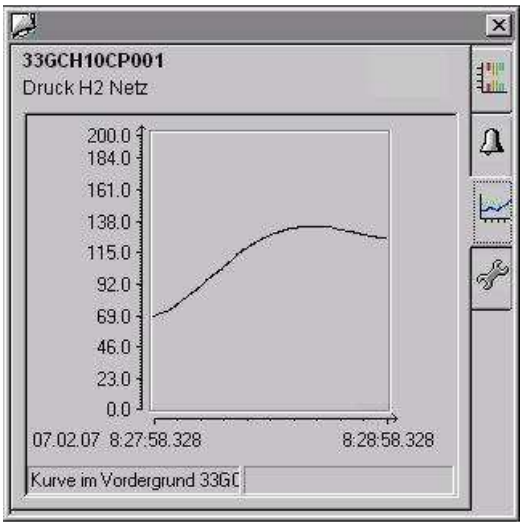
Bei Verletzung eines Grenzwertes wird dies mit dem Entsprechenden Farbumschlag im Button **"OG2"** (Rot -> Alarm) bzw. **"OG1"** (Gelb -> Warnung) dargestellt.
Die Wertigkeit der Grenzwerte wird neben den entsprechenden Buttons angezeigt.

Es können bis zu 2 untere Grenzwerte definiert sein. Diese Grenzwerte können über die Button **"UG2"** bzw. "**UG1"** im Faceplate verändert werden. Die Grenzwerte können nur mit der entsprechenden Bedienberechtigung verändert werden!

Bei Verletzung eines Grenzwertes wird dies mit dem Entsprechenden Farbumschlag im Button **"UG2"** (Rot -> Alarm) bzw. **"UG1"** (Gelb -> Warnung) dargestellt.
Die Wertigkeit der Grenzwerte wird neben den entsprechenden Buttons angezeigt.

Die Istwertanzeige zeigt den aktuellen skalierten Messwert an.
Über den Button "**XINB**" kann bei entsprechender Bedienberechtigungsstufe das „Istwert Forcen“ aktiviert werden. Bei aktivem „Istwert-Forcen“ kann über den Button „**X**“ ein Istwert vorgegeben werden.

15.3.1 Anzeige Istwertgraph



Unter dem Karteireiter Graph wird die Kurzzeitarchivierung des Messwertes als Kurve dargestellt.

15.4 Analogausgabe

Der Baustein dient zur Ausgabe, zum Beobachten und Bedienen eines Analogwertes.
Störungen des Ausgangssignals werden alarmiert. Der Ausgabewert wird auf die Bereichsgrenzen normiert. Mit dem Baustein kann ein Regelventil angesteuert und seine Stellungsrückmeldung überwacht werden.

Im Statusfenster des Faceplates wird der aktuelle Ausgabewert in Balkenform, mit der im Baustein parametrierten Skalierung, dargstellt. Ebenso wird der Ausgabewert als Zahlenwert “Y“ = dargestellt.
Am Balkenanfang und Balkenende stehen jeweils der parametrierte Messbereichsanfang und das Messbereichsende.

Wird die Analogausgabe im Handbetrieb eingeschaltet erscheint ein Eingabebutton “**Y =**“. Hier kann nun ein Wert für die Ausgabe vorgegeben werden.

Befindet sich die Analogausgabe im Automatikbetrieb wird der Stellwert ausgegeben der unter “ **Yext =** “ eingetragen ist.

Ist die Analogausgabe ausgeschaltet wird der Stellwert ausgegeben der unter “ **YAUS =** “ eingetragen ist.

Neben dem Balken wird die Stellungsrückmeldung Ventil ZU “**RCDA**“ als blauer Text, mit aktuellem Zustand (0 oder 1), angezeigt.

15.5 Motor

Der Baustein dient zur Bedienung und Überwachung eines Motors mit einer Drehzahl. Er unterstützt die Betriebsarten Automatik- und Handbetrieb Es werden die Rückmeldungen vom Hauptschütz des Motors und des Motorschutzes überwacht.

Im Statusfenster wird oben links der aktuelle Zustand des Motors über eine Anzeigefenstergruppe dargestellt. Die Statusanzeige erfolgt mit Hilfe eines Farbumschlages des entsprechenden Anzeigefensters und einer Klartextanzeige unterhalb der Anzeigefenstergruppe.

In den unten gezeigten Beispielen ist der Motor eingeschaltet bzw. gestört.

Anstehende Störungen und Meldungen werden auf der rechten Seite des Statusfensters als Klartextmeldung angezeigt.

In diesem Beispiel sind einige Störungen und Meldungen dargestellt.

Unten links wird die Rückmeldung des Hauptschützes Ein “**RCAA**“ als blauer Text, mit aktuellem Zustand (0 oder 1), angezeigt.

15.6 Ventil

Der Baustein dient zur Bedienung und Überwachung eines Ventils mit einer Ansteuerung und zwei Rückmeldungen der Endlagen. Er unterstützt die Betriebsarten Automatik- und Handbetrieb Die Rückmeldungen der Endlagen werden auf Plausibilität überprüft.

Im Statusfenster wird oben links der aktuelle Zustand des Ventils über eine Anzeigefenstergruppe dargestellt. Die Statusanzeige erfolgt mit Hilfe eines Farbumschlages des entsprechenden Anzeigefensters und einer Klartextanzeige unterhalb der Anzeigefenstergruppe.

In den unten gezeigten Beispielen ist das Ventil eingeschaltet bzw. gestört.

Anstehende Störungen und Meldungen werden auf der rechten Seite des Statusfensters als Klartextmeldung angezeigt.

In diesem Beispiel sind einige Störungen und Meldungen dargestellt.

Unten links werden die Rückmeldungen der Endlagen AUF „“**RCAA**“ bzw. ZU “**RCDA**“ als blauer Text, mit aktuellem Zustand (0 oder 1), angezeigt.

15.7 Regler

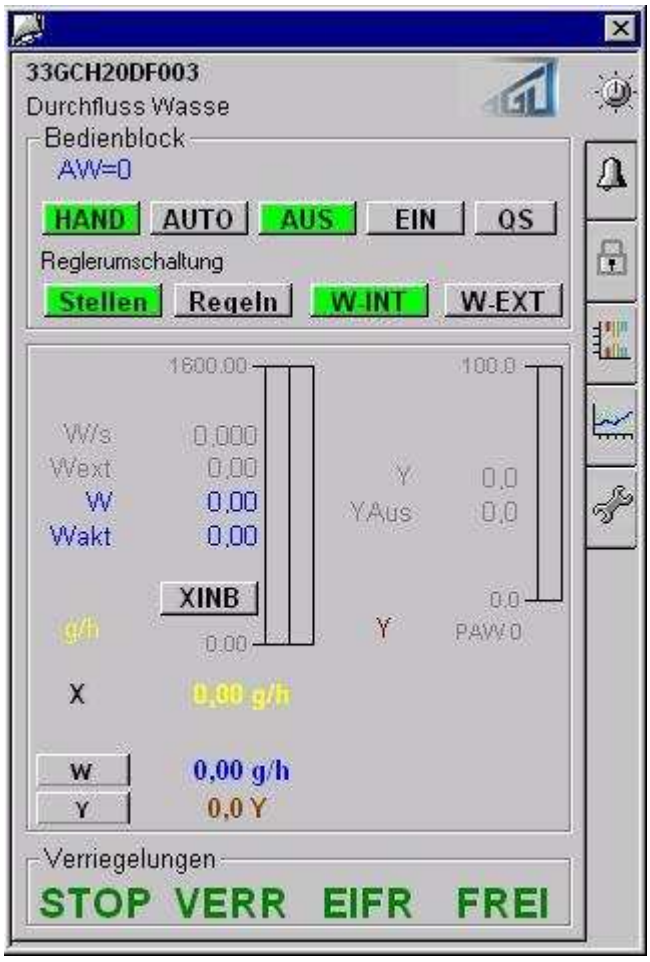
Der Baustein ist die Kombination aus einem Analogeingangs-, einem Analogausgangsbaustein und dem Baustein für den Regelalgorithmus.

Der Analogeingangsbaustein dient zum Einlesen, Beobachten und Überwachen eines Analogwertes. Es können vier Grenzwerte parametriert sein, die schaltend und oder alarmierend wirken. Störungen des Eingangssignals werden alarmiert. Der eingelesene Wert wird auf die Bereichsgrenzen skaliert.

Der Analogausgangsbaustein dient zum Ausgeben, Beobachten und Bedienen eines Analogwertes. Störungen des Ausgangssignals werden alarmiert. Der Ausgabewert wird auf die Bereichsgrenzen normiert. Mit dem Baustein kann ein Regelventil angesteuert und seine Stellungsrückmeldung überwacht werden.

Der Regelalgorithmus ist ein PID-Regler, der mittels der Parameter Kp, Tn und Tv eingestellt werden kann.

15.7.1 Bedienblock



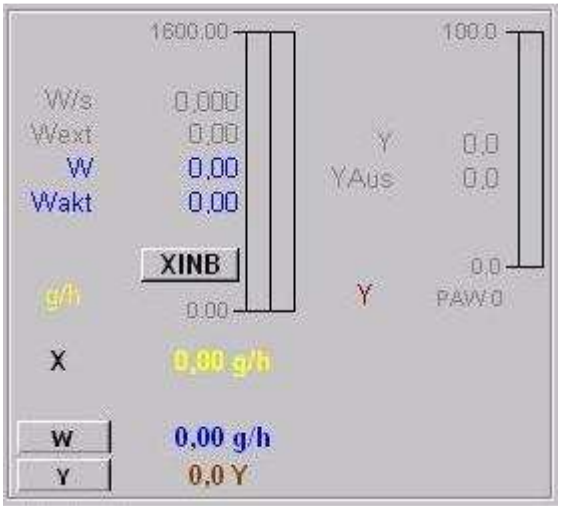
Der Bedienblock des Regers ist um einige Funktionen erweitert. Unter Reglerumschaltung kann der Regler auf "**Stellen**" bzw. "**Regeln**" umgeschaltet werden.

In der Betriebsart Stellen wird bei eingeschaltetem Regler, die Stellgröße ausgegeben, die vom Bediener unter "**Y**" eingegeben wurde.

In der Betriebsart Regeln wird bei eingeschaltetem Regler die Stellgröße vom Regelalgorithmus bestimmt.

Mit den Button "**W-INT**" bzw. "**W-EXT**" kann vorgewählt werden ob der Regelalgorithmus mit einem Sollwert von extern (z.B. über eine Schrittkette) oder von intern (manuelle Vorgabe über Faceplate) arbeiten soll.

15.7.2 Statusfenster



Im Statusfenster wird der aktuelle Sollwert als blauer Balken mit Skalierung angezeigt. Die Skalierung entspricht der des Istwertes.
Ebenso werden in Textform folgende Sollwerteinträge angezeigt:

„**W/s**": Verstellinkrement für die Sollwertrampe. Bei einem Wert von z.B. 0.2 wird die Änderung des Sollwertes über eine Rampe mit 0,2 Dimension pro Sekunde angefahren.

„**Wext**": Sollwert in der Betriebsart Automatik.

„**W**": Sollwert in der Betriebsart Hand.

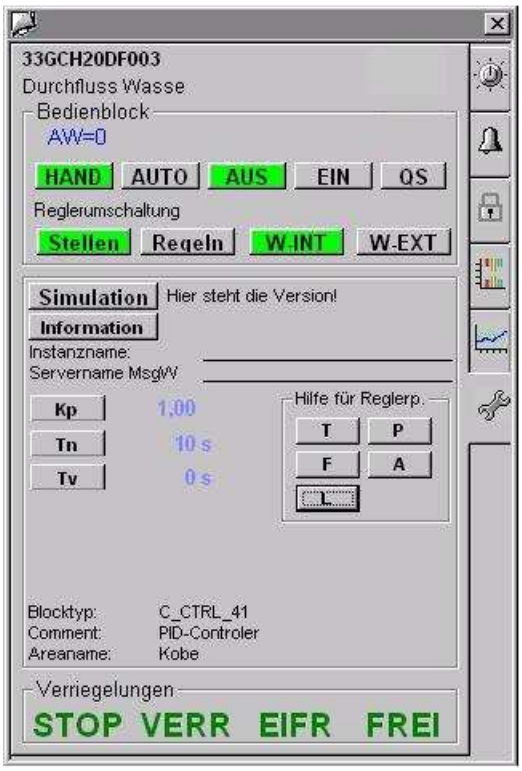
„**Wakt**": aktuell wirkender Sollwert

Der aktuelle Istwert wird zum einen als gelber Balken mit Skalierung und zum anderen als gelber Zahlenwert "**X**" angezeigt.

Über den Button "**XINB**" kann bei entsprechender Bedienberechtigungsstufe das „Istwert Forcen" aktiviert werden. Bei aktivem „Istwert-Forcen" kann über den Button „**X**" ein Istwert vorgegeben werden.
Eine reine Ansicht des Istwertes kann über den Karteireiter Istwert angewählt werden.

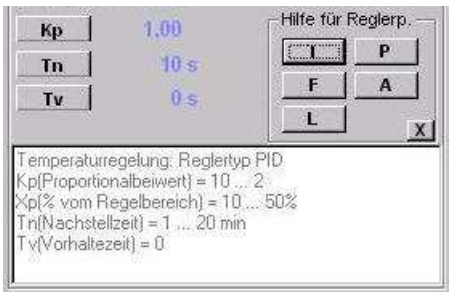
Der aktuelle Stellwert wird zum einen als brauner Balken mit Skalierung und zum anderen als brauner Zahlenwert "**Y**" angezeigt.
Ebenso wird neben dem Balken der "**Y Aus**" Parameter angezeigt. Der Parameter "**Y Aus**" gibt die Stellgröße vor, die bei ausgeschaltetem Regler ausgegeben wird.

15.7.3 Servicefenster



Im Karteireiter Servicefenster kann mit der entsprechenden Bedienberechtigungsstufe der Regler parametriert werden. Hier können über die bekannten Bediendialoge die Werte “**Kp**“, “**Tn**“ sowie “**Tv**“ vorgegeben werden.

Unter dem Eintrag “**Hilfe für Reglerparametrierung**“ können einige Standardparametrierungen für bestimmte Regleranwendungen abgerufen werden.



Folgende Regleranwendungen werden beschrieben:

- “**T**“ Temperaturregelung
- “**F**“ Flussregelung
- “**L**“ Höhenstandregelung
- “**P**“ Druckregelung
- “**A**“ Analyseregelung

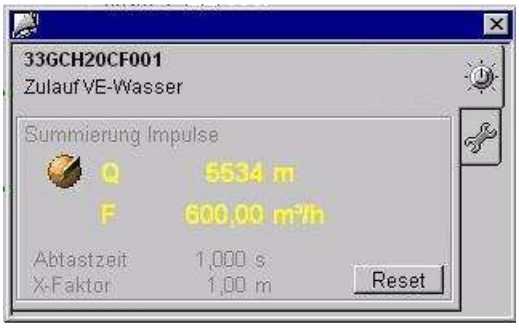
ACHTUNG !

Die unter Hilfe für Reglerparametrierung stehenden Einträge verstehen sich als Vorschläge.

15.8 Zähler

Der Baustein dient zur Berechnung eines Durchflusses (Summierung Impulse) aus einer mittels Impulse erfassten Menge. Die Impulse müssen dem Bausstein in Form eines Umlaufzählers zugeführt werden.

Zudem dient der Baustein zur Berechnung einer Menge (Integration Fluss) aus einem mittels Analogwert erfassten Durchfluss. Der Fluss muss dem Baustein in Form eines Realwerts zugeführt werden.



Im Stausfenster wird oben links die Betriebsart des Bausteines als Klartext angezeigt:

Summierung Impulse: Impulse werden summiert, der Fluss F wird berechnet.

Integration Fluss: Der Fluss wird gemessen und die Menge Q berechnet (Integration).

Das Symbol „Schloss" zeigt an, ob die Summierung innerhalb der AS freigegeben ist. Wird das Symbol dargestellt, ist die Summierung gesperrt.
Das Symbol „Sonnenuhr" zeigt an, ob die Summierung innerhalb der AS freigegeben ist. Wird das Symbol dargestellt, ist die Summierung freigegeben.

Der aktuelle Fluss und die aus dem Fluss aufsummierte Menge werden als Zahlenwert angezeigt.

Ebenso sind die Parameter Abtastzeit und X-Faktor dargestellt:

Abtastzeit: parametrierte Abtastzeit des Summierers in der Betriebsart „Integration Fluss"
X-Faktor: Wertigkeit eines Impulses in der Betriebsart „Summierung Impulse"

Der Button „**Reset**" ermöglicht das Rücksetzen des Zählwertes.

16 Erstellung von Grundfunktionen

Eine ausführliche Beschreibung der Erstellung von Grundfunktionen befindet sich in der Beschreibung des FW_Control Bausteins

17 WinCC flexible

Die Vorort Bedienung erfolgt mittels OP oder TP und dem Programmpaket SIMATIC WinCC flexible.

Systemvoraussetzung

Der Standard unterstützt die Projektierung der Panels „WinCCflexible 2007“.

Beim Einsatz eines Siemens TP oder MP, muss dieses als „integriertes“ Projekt innerhalb des Simatic Managers bearbeitet werden. Die Datenzugriffe werden direkt auf das S7 Projekt verknüpft.

Anmerkung:
Um WinCC flexible in STEP 7 zu integrieren, beachten Sie die Reihenfolge bei der Installation. Installieren Sie zuerst die STEP 7 Software und anschließend WinCC flexible. Die Installationsroutine von WinCC flexible erkennt eine vorhandene STEP 7 Installation und installiert automatisch die Unterstützung für die Integration in STEP 7.

Hardwarevoraussetzungen

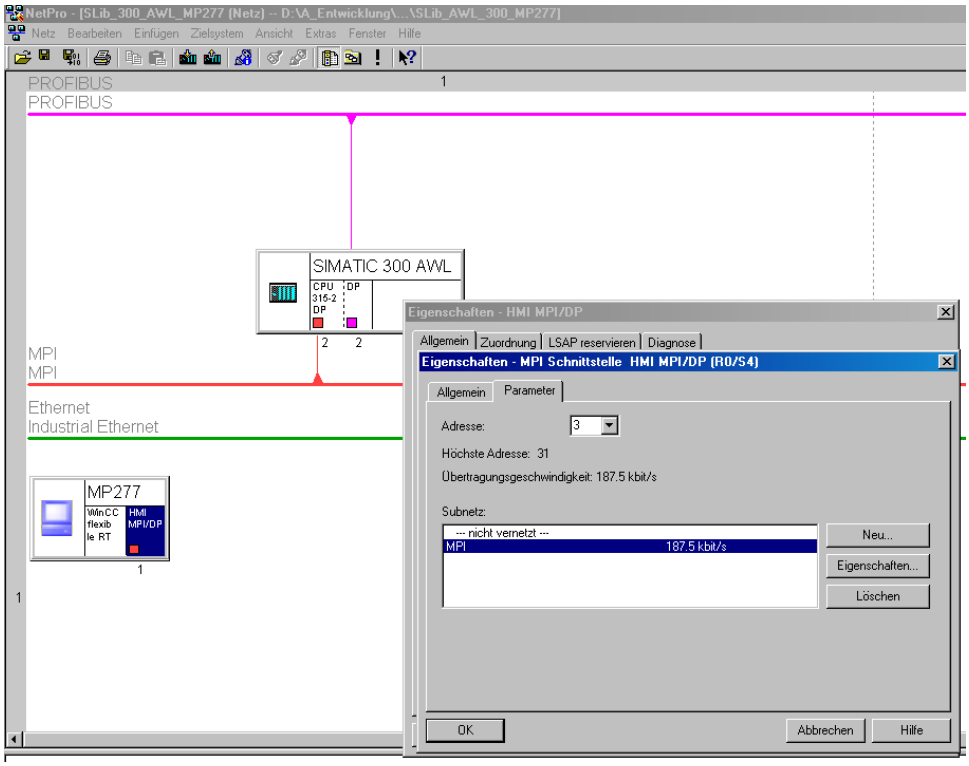
Das Panel ist entsprechend dem erforderlichen Mengengerüst auszuwählen.
Hierbei ist z.B auf die Displaygröße, die Anzahl Variablen, die Anzahl Skripte, die Farbdarstellung usw. zu achten.

17.1 SmartLib WinCC flexible Option zu einem Projekt hinzufügen

Entpacken Sie die Datei „Version SmartLibWinCCflexible_4_1_X_X.zip“. Öffnen Sie die SmartLib WinCCflexible Option „SmartLibWinCCflexible_4_1_X_X.hmi“ im Ordner „SmartLibWinCCflexible_4_1_X_X“ mit WinCCflexible. Integrieren Sie dieses Projekt in ihr bestehendes SmartLib-S7-Projekt über den Menüpunkt „Projekt/In Step 7 integrieren“. In ihrem Step7-Projekt wird jetzt ein WinCCflexible Projekt mit der SmartLib-Option integriert. Die folgende Abbildung zeigt exemplarisch ein Step7-Projekt mit integriertem WinCCflexible Projekt (MP277) im Simatic-Manager.

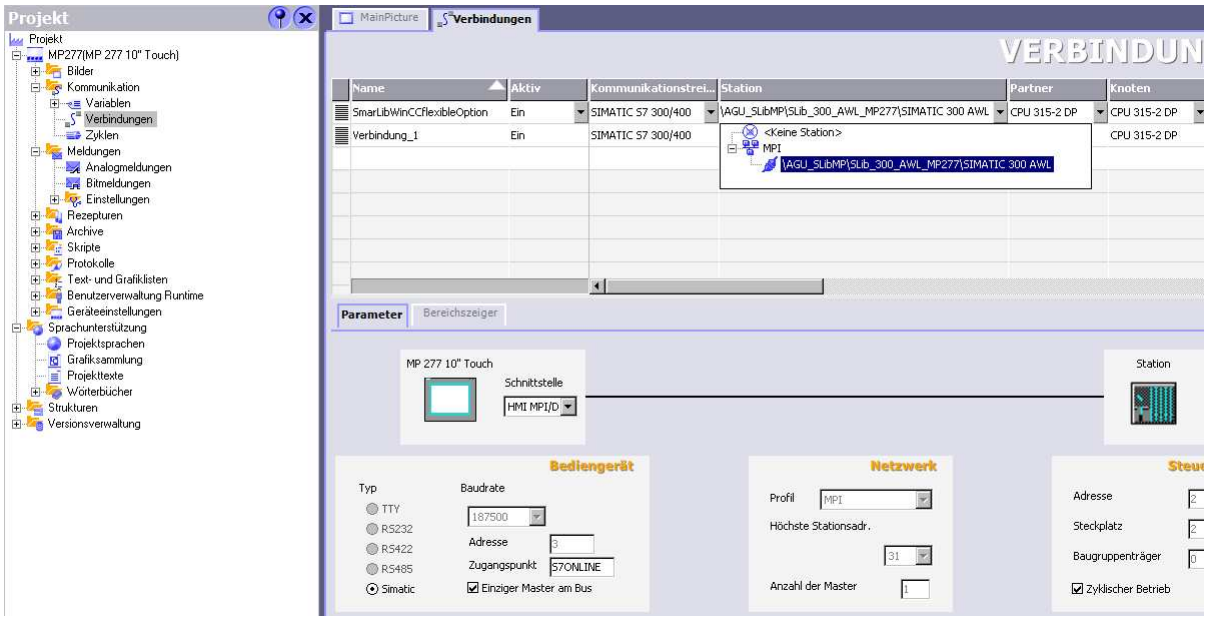
17.2 Verbindung projektieren

Für den Datenaustausch zwischen dem Bediengerät und der Steuerung ist eine Verbindung zu projektieren. Mit NetPro projektieren sie die Verbindung auf einer grafisch unterstützten Oberfläche. Durch die Integration in das Step7-Projekt wurde in NetPro das MP277 eingetragen. Hier ist eine Adresse zu projektieren und die gewünschte Kommunikation (MPI/Profibus) auszuwählen. Abschließend muss die Konfiguration abgespeichert und übersetzt werden.



Die geänderte Hardwarekonfiguration ist noch in das AG zu laden.

In WinCC flexible wird dadurch die „Verbindung_1“ angelegt. Ordnen Sie der WinCC flexible Verbindung „SmartLibWinCCflexibleOption“ die Station der „Verbindung_1“ zu und löschen anschließend die „Verbindung_1“.

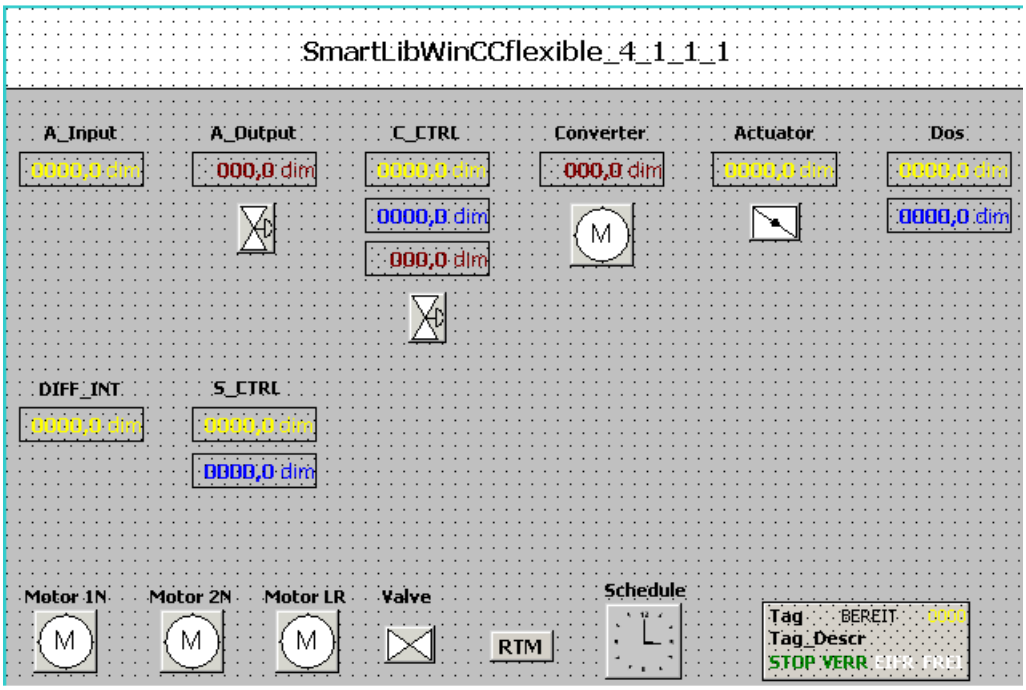


Abschließend sollte das WinCC flexible – Projekt gespeichert und neu generiert werden.

17.3 Bilder

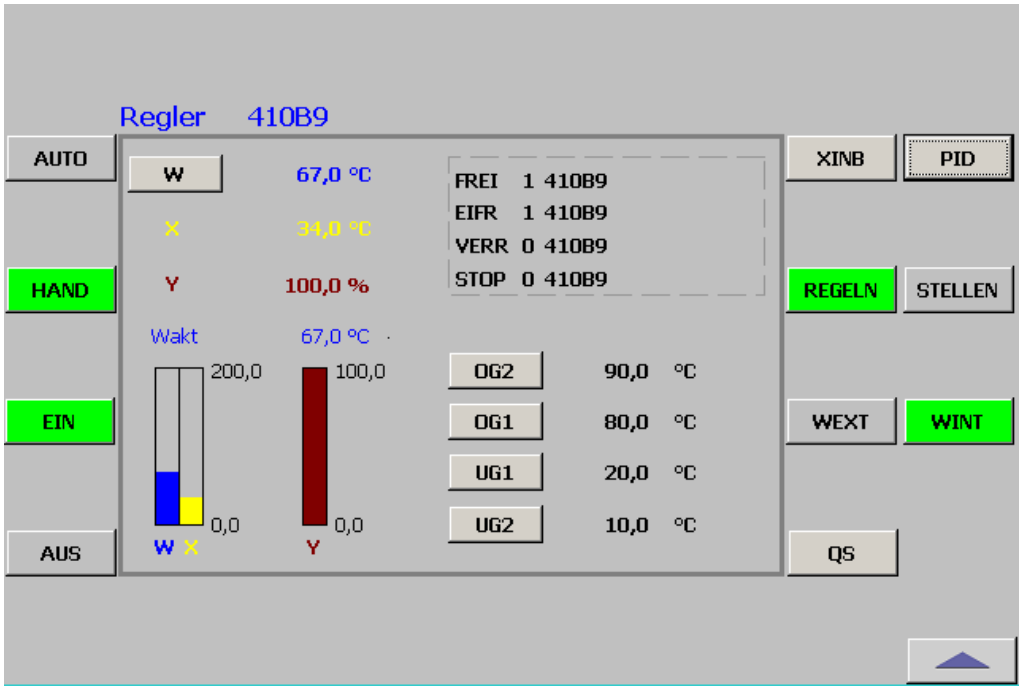
17.3.1 Prozessbild

Zum Erstellen der Fliessbilder (Prozessbilder/Anlagenbilder) können die benötigten Fliessbildsymbole aus dem Bild „@AGU_SmartLib_WinCCflexible_FB_Symbole“ kopiert werden.



17.3.2 Kreisbild

In WinCC flexible gibt es zu jedem Sm@rtLib Bausteintyp, sofern erforderlich, ein WinCC flexible-Kreisbild. Nachfolgend ist exemplarisch ein Kreisbild für den C_CTRL-Baustein (Regler) dargestellt.



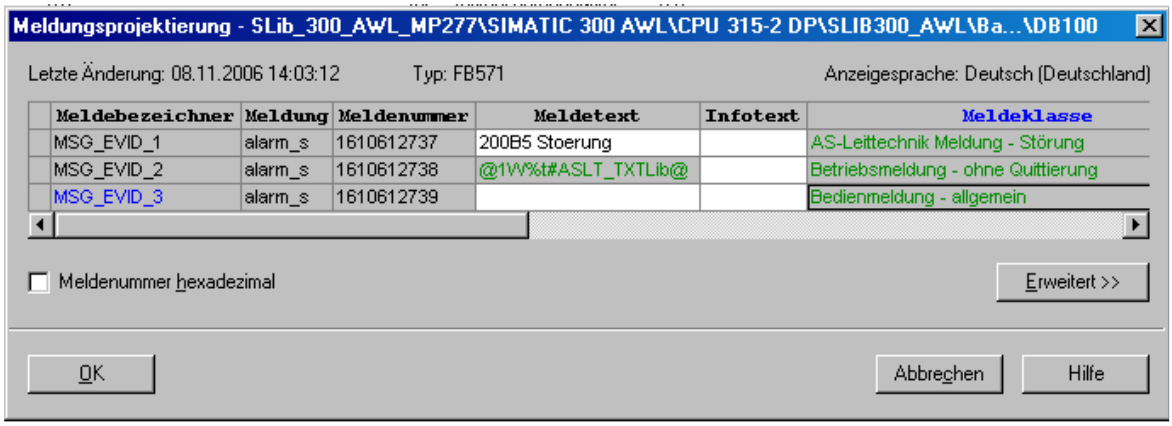
17.3.3 Systembild

Das Systembild sollte die folgenden Funktionen bereitstellen:

An/Abmelden
Uhrzeit anzeigen
Transfer
Runtime beenden
Reinigung Touchscreen
Kalibrierung Touchscreen

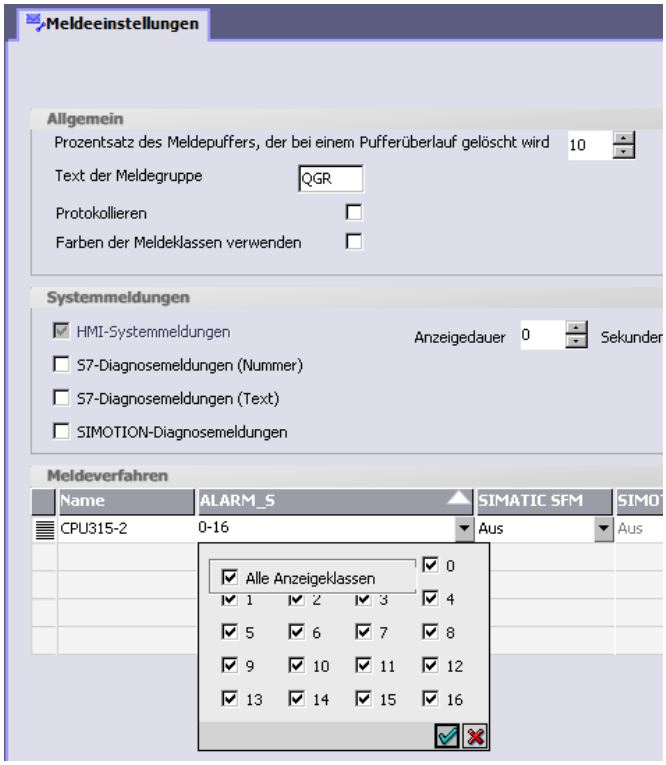
17.4 Meldesystem

Die SmartLib-Bausteine erzeugen Meldungen mittels eines Alarm_S-Bausteins.
Der Alarm_S-Baustein arbeitet mittels des Meldenummernverfahrens. Die Meldenummern werden automatisch bei der Projektierung in STEP 7 vergeben. Anhand der Nummern erfolgt die eindeutige Zuordnung der Meldetexte. Für jede MSR-Stelle (Instanz) muss ein Meldetext projektiert werden. Der Dialog Meldungsprojektierung wird über das Kontextmenü des jeweiligen Instanz-DB „Spezielle Objekteigenschaften/Meldung...“ aufgerufen.



Bei der Meldungsprojektierung in STEP 7 werden die hinterlegten Texte und Attribute in den Projektierungsdaten von STEP 7 abgelegt. WinCC flexible importiert automatisch die erforderlichen Daten und überträgt sie später beim Transfer auf das Bediengerät.

In WinCC flexible können Sie die Ausgabe der ALARM_S-Meldungen über Anzeige-Klassen filtern.

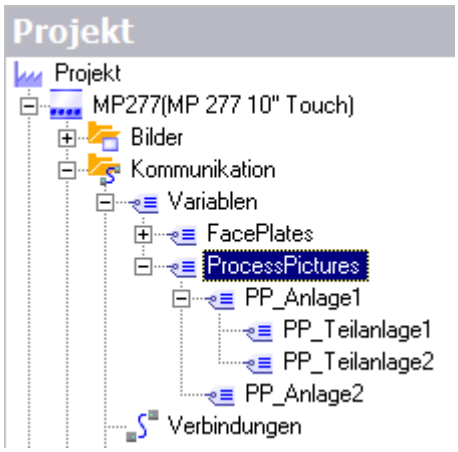


Wählen Sie im Projektfenster "Meldungen -> Einstellungen" und doppelklicken Sie "Meldeeinstellungen". Im Bereich "Meldeverfahren" werden die bestehenden Verbindungen angezeigt.
Selektieren Sie in der Zeile der gewünschten Verbindung das Feld in der Spalte "ALARM_S" und öffnen Sie über die Auswahlschaltfläche den Auswahldialog.

Wählen Sie „Alle Anzeigeklassen“ aus. Schließen Sie den Auswahldialog mit der Schaltfläche .

17.5 Start der Projektierung und Voraussetzungen

Vorraussetzung: Innerhalb des STEP7-Programms wurden die Sm@rtLib-Funktionsbausteine mit den Instanz DB’s angelegt. Wir empfehlen die Variablen im WinCC flexible-Projekt eindeutig nach Anlage/Teilanlage abzulegen. Dazu kann in WinCC flexible eine Struktur mit Ordnern angelegt werden.



17.5.1 Fliessbild erstellen

Erstellen Sie bitte die Fliessbilder (Process Picture (PP)) unter Verwendung der Fliessbildsymbole aus dem Bild „@AGU_SmartLib_WinCCflexible_FB_Symbole“

17.5.2 Fliessbildvariablen anlegen

Projektieren Sie für jede Instanz in einem Fliessbild die Variablen entsprechend der u.a. Tabelle. Die Variablennamen sollten zur besseren Unterscheidung die MSR-Stellen-Bezeichnung (AKZ) enthalten. Für den SCHEDULE- und CLOCKTIMER-Baustein sind keine Variablen erforderlich.

TYP	NAME (Aktor/Sensor)	Kennzeichnung	Format	Ort	Datenquelle
A_INPUT (Analogeingang)	PP_AKZ_	Q_PV_LC	REAL	Instanz DB	DBD70
		Q_STATE	DWORD		DBD120
A_OUTPUT (Analogausgang)	PP_AKZ_	Q_MV	REAL	Instanz DB	DBD68
		Q_STATE	DWORD		DBD96
		Q_OP_STATE	INT		DBW92
C_CTRL (Regler)	PP_AKZ_	Q_PV	REAL	Instanz DB	DBD180
		Q_SP_ACTIVE	REAL		DBD208
		Q_MV	REAL		DBD212
		Q_AI_STATE	DWORD		DBD236
		Q_STATE	DWORD		DBD232
		Q_MV_STATE	INT		DBW230
CONVERTER (Umrichter)	PP_AKZ_	Q_MV	REAL	Instanz DB	DBD92
		Q_STATE	DWORD		DBD118
		Q_OP_STATE	INT		DBW116
ACTUATOR (Stellantrieb)	PP_AKZ_	Q_PV_LC	REAL	Instanz DB	DBD106
		Q_STATE	DWORD		DBD114
		Q_OP_STATE	INT		DBW144
DOS (Dosierer)	PP_AKZ_	Q_PV_QUANT	REAL	Instanz DB	DBD98
		Q_SP	REAL		DBD106
		Q_STATE	DWORD		DBD136
DIFF_INT (Summierer)	PP_AKZ_	Q_PV_QUANT	REAL	Instanz DB	DBD40
MOTOR _1N (Motor 1 Drehrichtung)	PP_AKZ_	Q_OP_STATE	INT	Instanz DB	DBW60
MOTOR _2N (Motor 2 Drehrichtung)	PP_AKZ_	Q_OP_STATE	INT	Instanz DB	DBW68
MOTOR_LR (Motor Links/Rechts)	PP_AKZ_	Q_OP_STATE	INT	Instanz DB	DBW64
VALVE (Ventil)	PP_AKZ_	Q_OP_STATE	INT	Instanz DB	DBW56
S_CTRL (Schritt-Regler)	PP_AKZ_	Q_PV	REAL	Instanz DB	DBD178
		Q_SP_ACTIVE	REAL		DBD206
		Q_AI_STATE	DWORD		DBD226
		Q_STATE	DWORD		DBD222
FW_CONTROL (Grundfunktion)	PP_AKZ_	Q_STEP	INT	Instanz DB	DBW90
		Q_STATE	DWORD		DBD114
		Q_HEAD_STATE	INT		DBW98

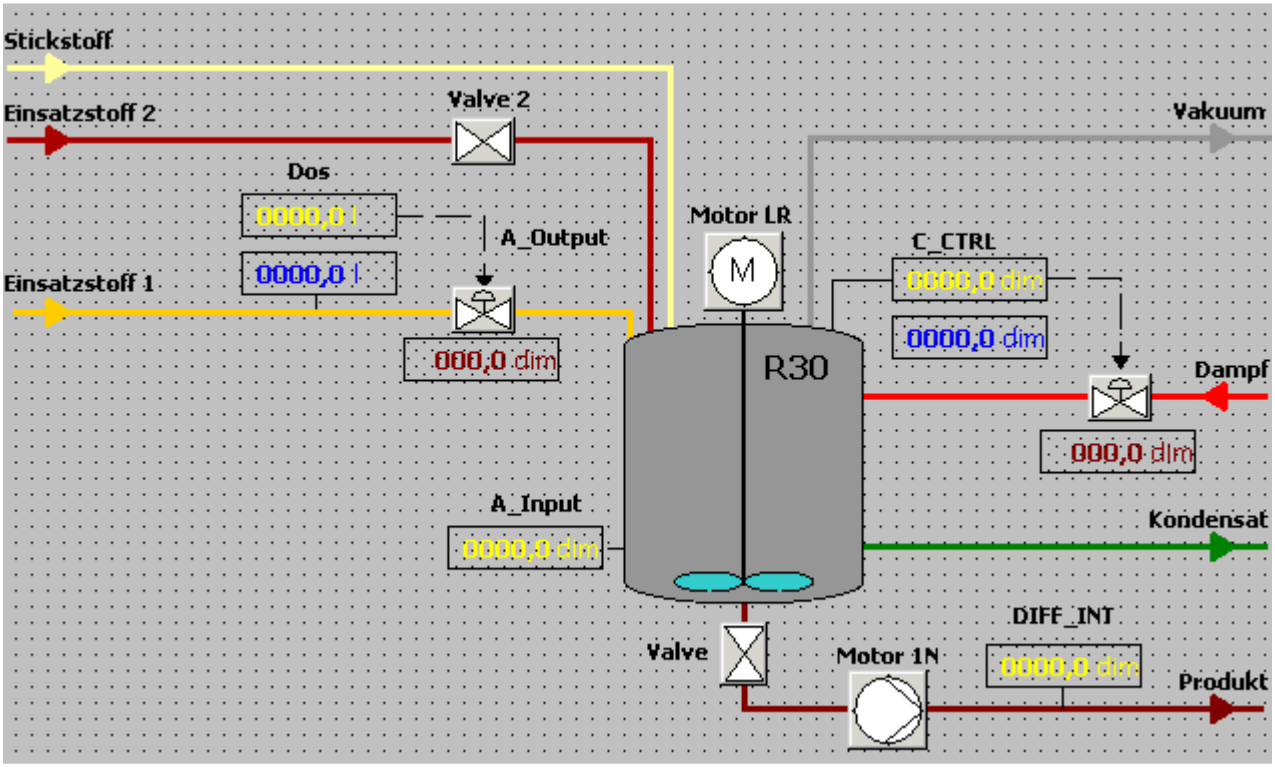
Als Beispiel sind für eine Temperaturmessung T001 folgende Fliessbildvariablen anzulegen:

PP_T001_Q_PV
PP_T001_Q_STATE

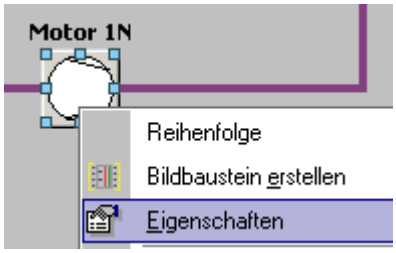
17.6 Visualisierung von Aktoren

17.6.1 Variablenverbindung

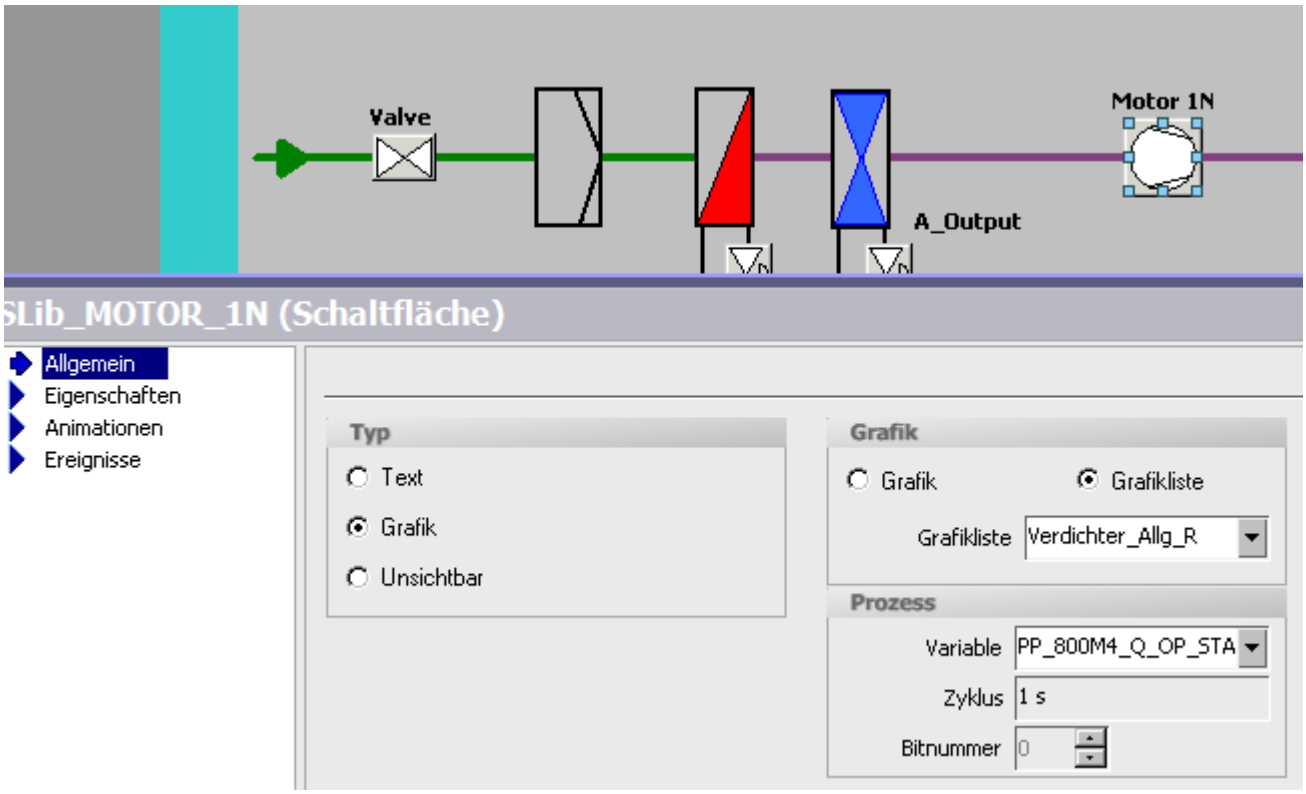
Bitte öffnen Sie das gewünschte Fliessbild. Das Bild öffnet sich und der gewünschte Aktor kann markiert werden. Nachfolgend ist examplarisch ein Fliessbild mit markiertem Aktor (Motor_1N) dargestellt.



Markieren Sie den gewünschten Aktor und öffnen die Eigenschaften.



Im unteren Bereich des Bildes sind nun die Eigenschaften des angewählten Elementes sichtbar.



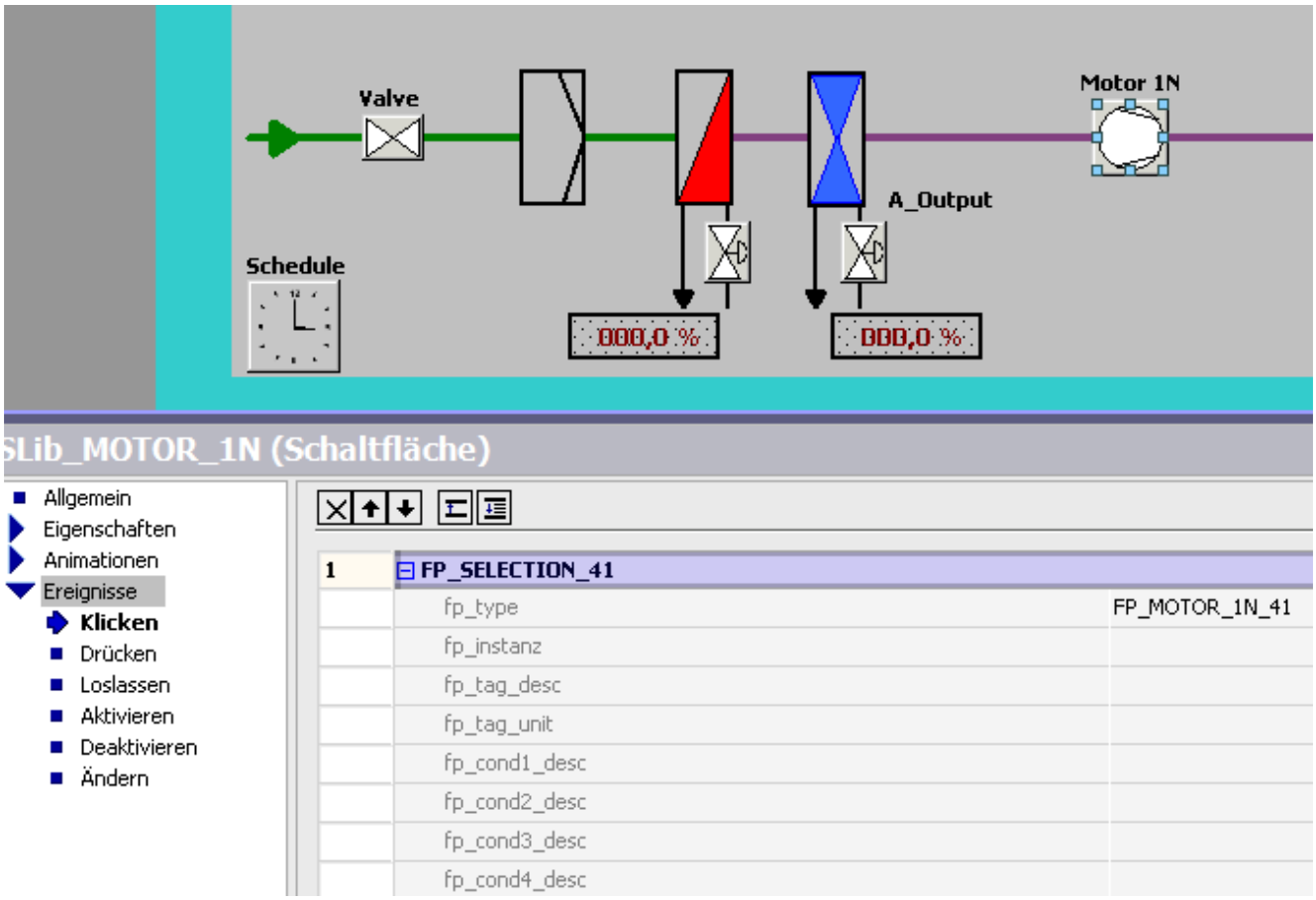
Wählen Sie im Eigenschaftsdialog unter „Allgemein“ folgendes:

- Im Feld Grafikliste/Grafikliste wählen Sie eine Grafikliste aus.
  Erstellen Sie eigene Grafiklisten, wenn die Vorlagen nicht ausreichen.

- Im Feld Prozess/Variable wählen Sie die instanzspezifische Variable aus.

17.6.2 Faceplate Anwahl dynamisieren

Klicken Sie das gewünschte Fliessbildsymbol des Aktors an und projektieren unter Ereignisse/Klicken die u.a. Eigenschaften.



Folgende Einstellungen müssen/können vorgenommen werden:

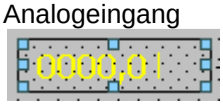
fp_type	Name des Bausteintyps (siehe Skript, FP_SELECTION_41)
fp_instanz	Nummer des Datenbausteins
fp_tag_desc	Überschrift des Faceplates: Messstellenbezeichnung (30 Zeichen)
fp_tag_unit	Einheit: Nur für Wertausgaben erforderlich : Einheit (6 Zeichen)
fp_cond1_desc	Bedingungstext 1 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond2_desc	Bedingungstext 2 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond3_desc	Bedingungstext 3 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond4_desc	Bedingungstext 4 (16 Zeichen) – sofern im S7-Programm genutzt

Der Aufruf der Faceplates (Bedienmasken für Handeingaben) ist aus den Anlagenfliessbildern durch einen Klick auf den Aktor möglich.

17.7 Visualisierung von Werten

Wertsymbole im Bild können Istwerte, Sollwerte oder Stellgrößen sein. Exemplarisch ist das Symbol für einen Analogeingang dargestellt.

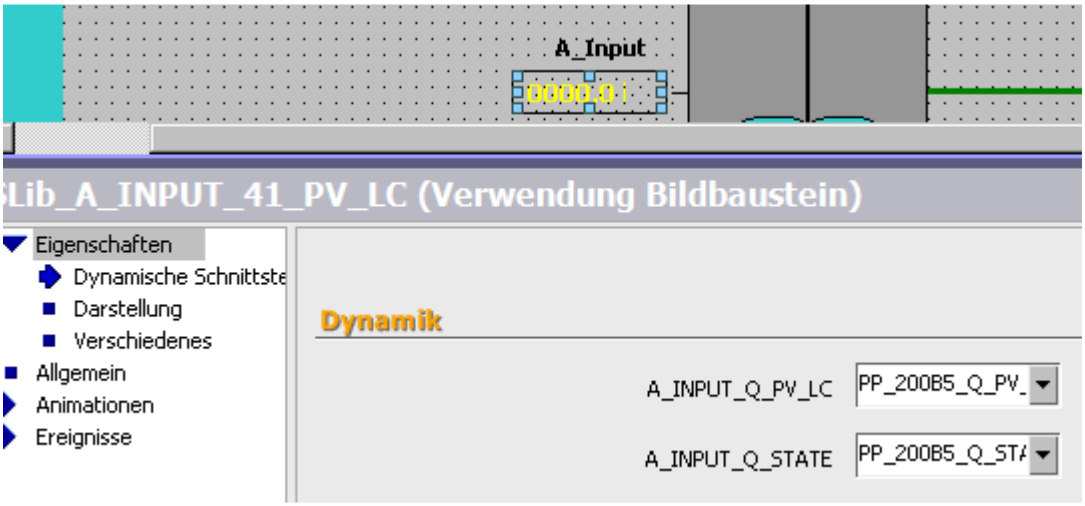
Analogeingang



17.7.1 Variablenverbindung

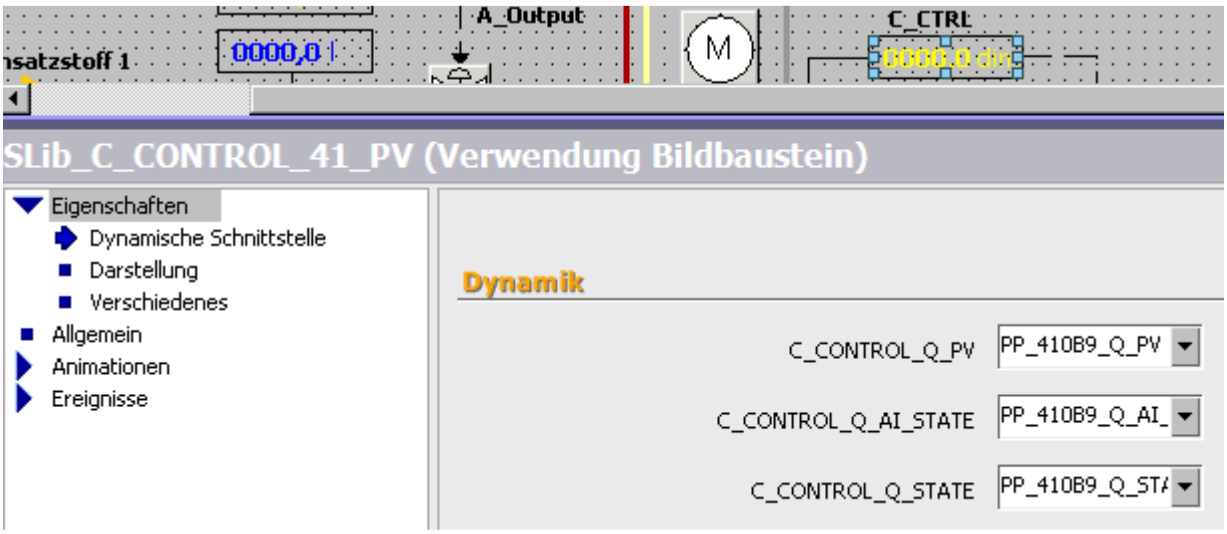
Ausgabewert dynamisieren:
Im Eigenschaftsdialog der Fliessbildwertsymbole sind unter Eigenschaften/Dynamische Schnittstelle/Dynamik die instanzspezifischen Variablen zu verschalten.

Dynamik	Variable	Beschreibung
A_INPUT_Q_PV_LC	PP_AKZ_Q_PV_LC	Istwert
A_OUTPUT_Q_MV	PP_AKZ_Q_MV	Stellwert
C_CTRL_Q_PV	PP_AKZ_Q_PV	Istwert
C_CTRL_Q__SP_ACTIVE	PP_AKZ_Q_SP_ACTIVE	aktiver Sollwert
C_CTRL_Q_MV	PP_AKZ_Q_MV	Stellwert
CONVERTER_Q_MV	PP_AKZ_Q_MV	Stellwert
ACTUATOR_Q_PV_LC	PP_AKZ_Q_PV_LC	Istwert
DOS_Q_SP	PP_AKZ_Q_SP	Sollwert
DOS_Q_PV_QUANT	PP_AKZ_Q_PV_QUANT	Istwert
DIFF_INT_Q_PV_QUANT	PP_AKZ_Q_PV_QUANT	Istwert
S_CTRL_Q_PV	PP_AKZ_Q_PV	Istwert
S_CTRL_Q__SP_ACTIVE	PP_AKZ_Q_SP_ACTIVE	aktiver Sollwert

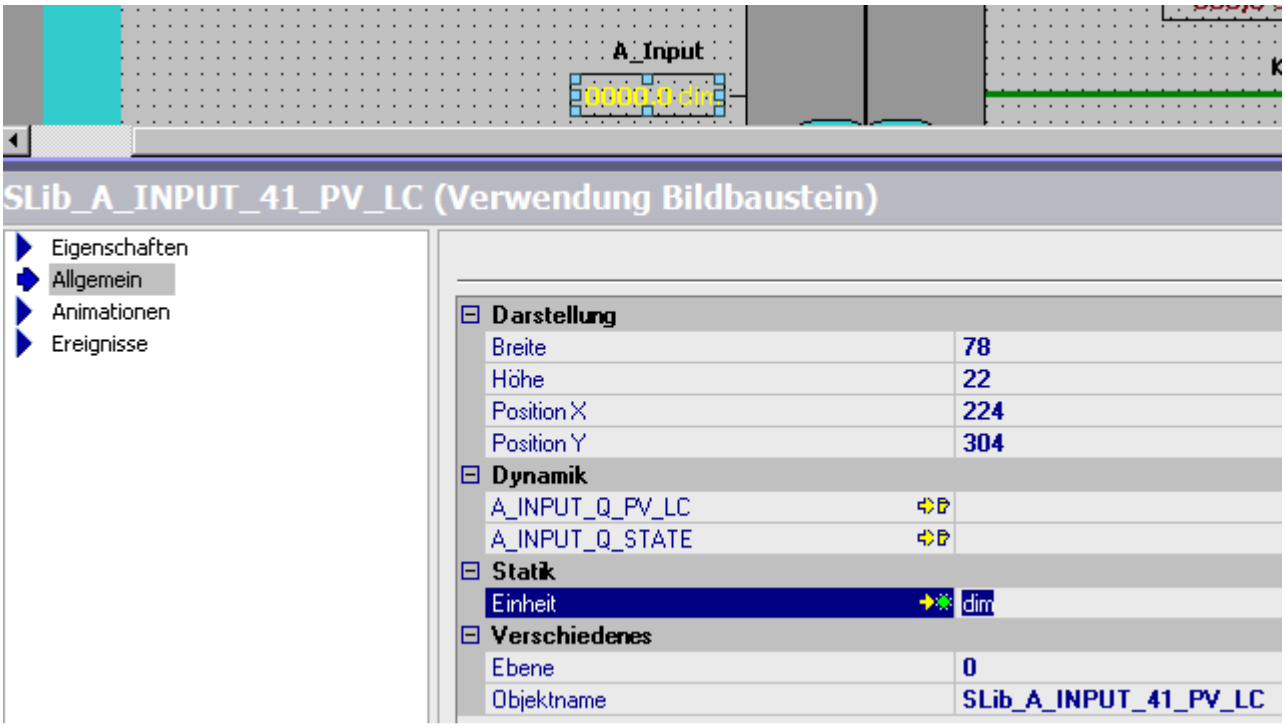


Zustände (Stati) dynamisieren:
Im Eigenschaftsdialog der Fliessbildwertsymbole sind unter Eigenschaften/Dynamische Schnittstelle/Dynamik die instanzspezifischen Variablen zu verschalten.

Dynamik	Variable	Beschreibung
A_INPUT_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
A_OUTPUT_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
C_CTRL_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
C_CTRL_Q_AI_STATE	PP_AKZ_Q_AI_STATE	Status für Visualisierung
CONVERTER_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
ACTUATOR_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
DOS_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
S_CTRL_Q_STATE	PP_AKZ_Q_STATE	Status für Visualisierung
S_CTRL_Q_AI_STATE	PP_AKZ_Q_AI_STATE	Status für Visualisierung

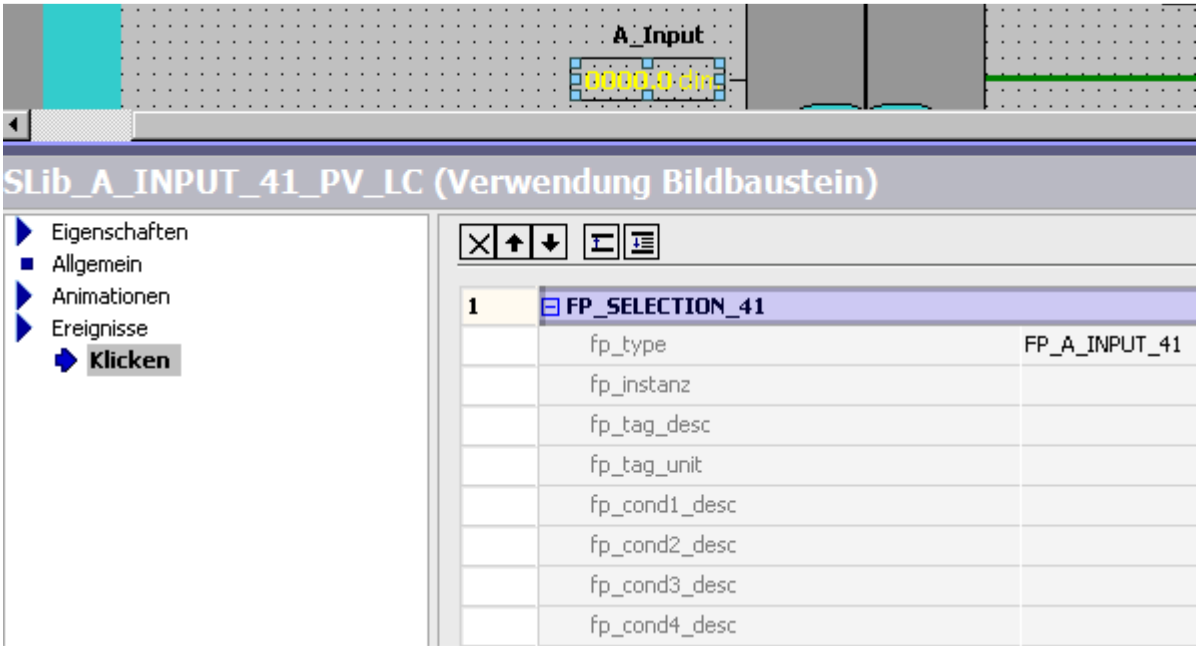


Einheit projektieren:
Im Eigenschaftsdialog der Fliessbildwertsymbole ist unter Allgemein/Statik/Einheit die gewünschte Einheit zu projektieren (Text).



17.7.2 Faceplate Anwahl dynamisieren

Klicken Sie das gewünschte Fliessbildsymbol des Aktors an und projektieren unter Ereignisse/Klicken die u.a. Eigenschaften.



Folgende Einstellungen müssen/können vorgenommen werden:

fp_type	Name des Bausteintyps (siehe Skript, FP_SELECTION)
fp_instanz	Nummer des Datenbausteins
fp_tag_desc	Überschrift des Faceplates: Meßstellenbezeichnung (30 Zeichen)
fp_tag_unit	Einheit: Nur für Wertausgaben erforderlich : Einheit (6 Zeichen)
fp_cond1_desc	Bedingungstext 1 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond2_desc	Bedingungstext 2 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond3_desc	Bedingungstext 3 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond4_desc	Bedingungstext 4 (16 Zeichen) – sofern im S7-Programm genutzt

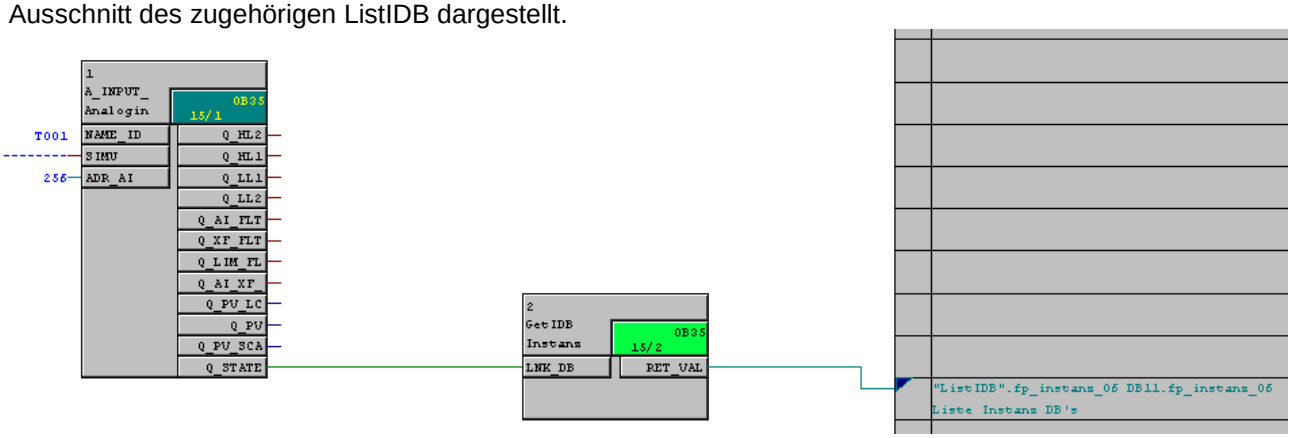
Der Aufruf der Faceplates (Bedienmasken für Handeingaben) ist aus den Anlagenfliessbildern durch einen Klick auf den Sensor möglich.

17.8 Besonderheiten bei der Projektierung mit CFC

Im gegensatz zu AWL, wo der Programmierer die Instanz-Datenbaustein-Nummern vorgibt, hat der Programmierer bei CFC keinen Einfluss auf die Instanz-Datenbaustein-Nummern der eingebauten Instanzen. Diese werden vom CFC-Editor beim Übersetzen des Programms automatisch vergeben. Für die Versorgung der Variablen in WinCC flexible müssen deshalb die vom Übersetzer vergebenen Instanz-Datenbaustein-Nummern für jede eingebaute Instanz ermittelt (GetIDB) und in einem Global-Datenbaustein (ListIDB) mit festen Adressen abgelegt werden. Für die folgenden Ausführungen wird für den ListIDB der DB11 verwendet.

17.8.1 GetIDB/ListIDB

Jede Instanz eines SmartLib-Funktionsbausteins muß im CFC-Plan mit einem GetIDB-Baustein verschaltet werden. Dieser Baustein ermittelt die Instanz-DB-Nummer und legt diese im ListIDB-Baustein ab. Die Organisation, an welcher Stelle im ListIDB die Instanz-DB-Nummern abgelegt werden, ist Aufgabe des Programmierers. Der GetIDB-Baustein kann an jeden beliebigen Ausgang des SmartLib-Baustein verschaltet werden, hier wurde der Ausgang Q_STATE gewählt. In der mitgelieferten Bibliothek „SlibWfle“ ist sowohl der GetIDB-Baustein als auch der ListIDB-Baustein enthalten. Diese beiden Bausteine müssen in den Bausteinordner des Step7-Projektes kopiert werden. Nachfolgend ist exemplarisch für eine Temperaturmessung T001 ein CFC-Plan und ein Ausschnitt des zugehörigen ListIDB dargestellt.



KOP/AWL/FUP - [DB11 -- "ListIDB" -- SLib_300_CFC\SIMATIC 300 CFC\CPU 315-2 DP\...\DB11]

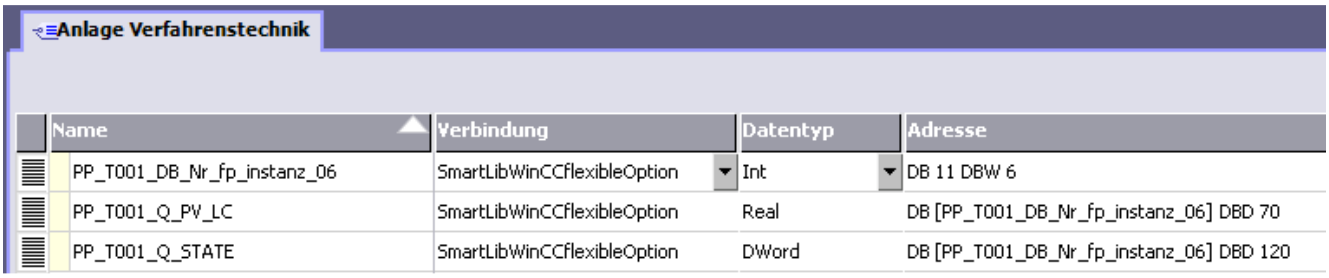
Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

Bibliotheken

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	Reserve	INT	100	100	Reserve
2.0	fp_instanz_02	INT	100	100	A_INPUT
4.0	fp_instanz_04	INT	100	100	
6.0	fp_instanz_06	INT	100	100	T001
8.0	fp_instanz_08	INT	100	100	
10.0	fp_instanz_10	INT	100	100	
12.0	fp_instanz_12	INT	100	100	
14.0	fp_instanz_14	INT	100	100	
16.0	fp_instanz_16	INT	100	100	
18.0	fp_instanz_18	INT	100	100	
20.0	fp_instanz_20	INT	100	100	
22.0	fp_instanz_22	INT	100	100	A_OUTPUT
24.0	fp_instanz_24	INT	100	100	

17.8.2 Fliessbildvariablen anlegen

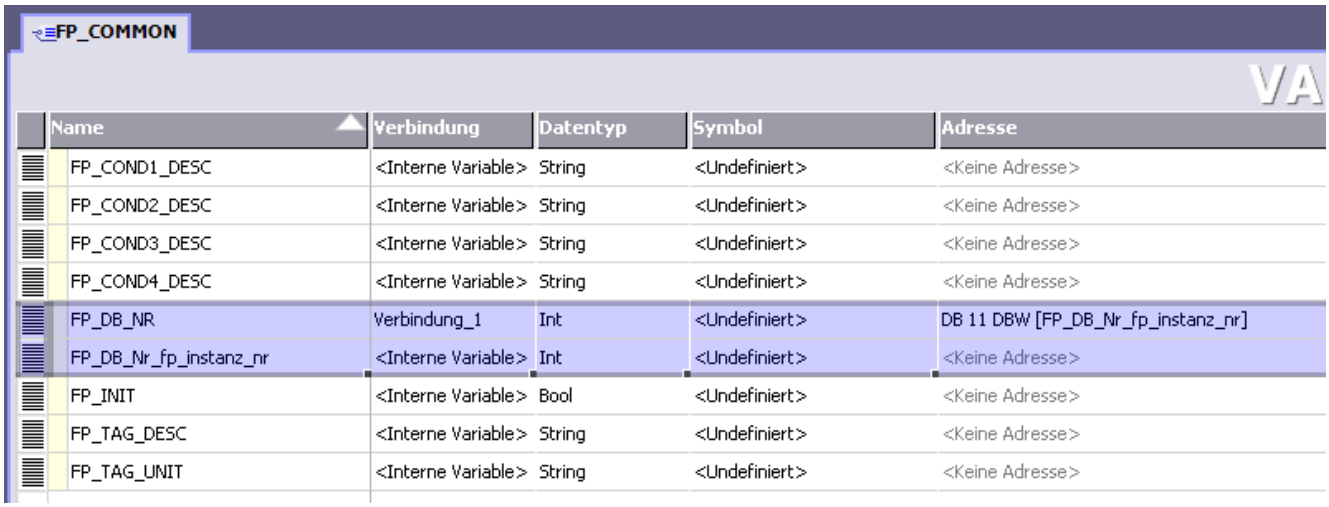
Für jede Instanz ist eine zusätzliche Hilfsvariable anzulegen. Für die Temperaturmessung T001 ist die Variable **PP_T001_DB_Nr_fp_instanz_06 anzulegen.** Diese ist erforderlich, um aus dem ListIDB die Daten-Baustein-Nummer für die Instanz T001 auszulesen. Die Adressen für die Fliessbildvariablen werden aus der Hilfsvariablen (Instanz-DB-Nummer) und der Datenquelle im Instanz-DB zusammengesetzt. Nachfolgend sind exemplarisch die für die Temperaturmessung T001 anzulegenden Variablen dargestellt.



Name	Verbindung	Datentyp	Adresse
PP_T001_DB_Nr_fp_instanz_06	SmartLibWinCCflexibleOption	Int	DB 11 DBW 6
PP_T001_Q_PV_LC	SmartLibWinCCflexibleOption	Real	DB [PP_T001_DB_Nr_fp_instanz_06] DBD 70
PP_T001_Q_STATE	SmartLibWinCCflexibleOption	DWord	DB [PP_T001_DB_Nr_fp_instanz_06] DBD 120

17.8.3 Faceplatevariable im Ordner FP_COMMON anpassen

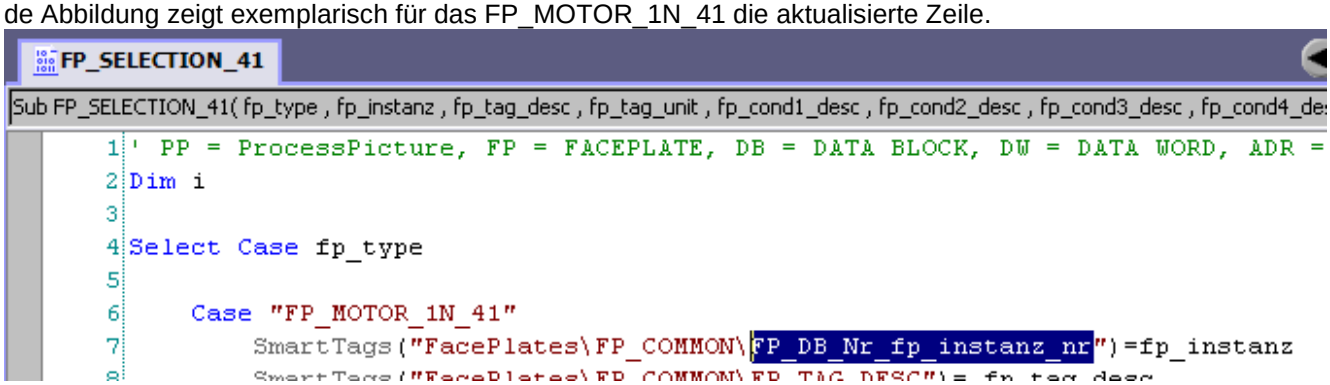
Die Variable **FP_DB_NR** ist für die Verwendung mit CFC wie in nachfolgender Abbildung dargestellt, anzupassen. Die Variable **FP_DB_Nr_fp_instanz_nr** ist bereits angelegt und darf nicht geändert werden.



Name	Verbindung	Datentyp	Symbol	Adresse
FP_COND1_DESC	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>
FP_COND2_DESC	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>
FP_COND3_DESC	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>
FP_COND4_DESC	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>
FP_DB_NR	Verbindung_1	Int	<Undefiniert>	DB 11 DBW [FP_DB_Nr_fp_instanz_nr]
FP_DB_Nr_fp_instanz_nr	<Interne Variable>	Int	<Undefiniert>	<Keine Adresse>
FP_INIT	<Interne Variable>	Bool	<Undefiniert>	<Keine Adresse>
FP_TAG_DESC	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>
FP_TAG_UNIT	<Interne Variable>	String	<Undefiniert>	<Keine Adresse>

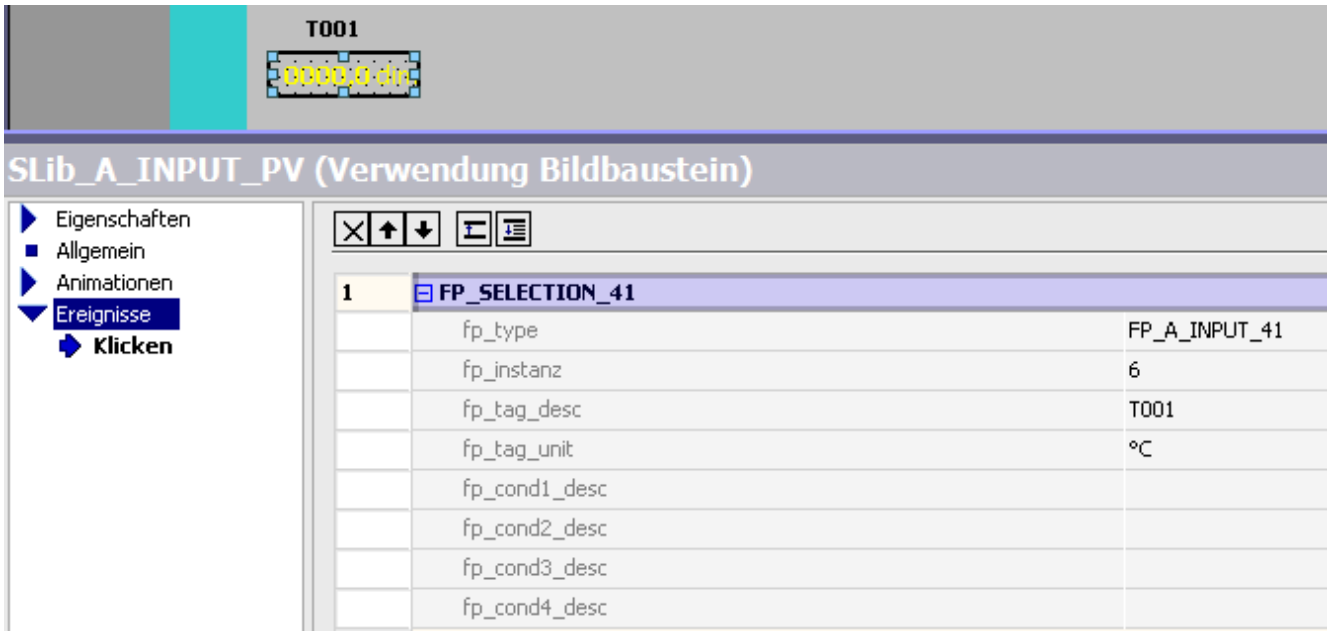
17.8.4 Faceplate Anwahlskript anpassen

Im Faceplateanwahl Skript **FP_SELECTION_41** ist die Zeichenfolge **FP_COMMON\FP_DB_NR** mittels Suchen/Ersetzen durch die Zeichenfolge **FP_COMMON\FP_DB_Nr_fp_instanz_nr** zu ersetzen. Die nachfolgende Abbildung zeigt exemplarisch für das FP_MOTOR_1N_41 die aktualisierte Zeile.


FP_SELECTION_41
Sub FP_SELECTION_41(fp_type , fp_instanz , fp_tag_desc , fp_tag_unit , fp_cond1_desc , fp_cond2_desc , fp_cond3_desc , fp_cond4_de
1 ' PP = ProcessPicture, FP = FACEPLATE, DB = DATA BLOCK, DW = DATA WORD, ADR =
2 Dim i
3
4 Select Case fp_type
5
6 Case "FP_MOTOR_1N_41"
7 SmartTags("FacePlates\FP_COMMON\FP_DB_Nr_fp_instanz_nr")=fp_instanz

17.8.5 Faceplate Anwahl dynamisieren

Klicken Sie das gewünschte Fliessbildsymbol des Aktors an und projektieren unter Ereignisse/Klicken die u.a. Eigenschaften. Bei fp_instanz ist hier die Adresse im ListIDB anzugeben, an der die ermittelte Instanz-DB-Nummer für die Temperatur T001 abgeleg wird.



Folgende Einstellungen müssen/können vorgenommen werden:

fp_type	Name des Bausteintyps (siehe Skript, FP_SELECTION)
fp_instanz	Adresse im ListIDB, wo mittels GetIDB die Datenbaustein-Nummmer abgelegt wird
fp_tag_desc	Überschrift des Faceplates: Meßstellenbezeichnung (30 Zeichen)
fp_tag_unit	Einheit: Nur für Wertausgaben erforderlich : Einheit (6 Zeichen)
fp_cond1_desc	Bedingungstext 1 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond2_desc	Bedingungstext 2 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond3_desc	Bedingungstext 3 (16 Zeichen) – sofern im S7-Programm genutzt
fp_cond4_desc	Bedingungstext 4 (16 Zeichen) – sofern im S7-Programm genutzt

Der Aufruf der Faceplates (Bedienmasken für Handeingaben) ist aus den Anlagenfliessbildern durch einen Klick auf den Sensor möglich.

18 Anhang

18.1 Programmerstellung mit CFC

Warum CFC

Ausgehend von einer ursprünglichen Forderung der Industrie, das SPS Programme auch von Personen aus dem Bereich der Betriebsbetreuung gelesen und erweitert werden, muss festgestellt werden, das die Werkzeuge der Programmerstellung den modernen Mitteln der SPS Automatisierung noch nicht konsequent angewendet werden. Zurzeit sind Spezialisten für die Erstellung und Pflege von AWL/FUP/KOP Programmen in vielen Fällen notwendig. Der Einsatz des Engineeringtools CFC (Continuus Funktion Chart) würde in vielen Fällen einen deutlichen Schritt hin zu mehr Transparenz und in der Software darstellen.

Auswirkung des CFC-Einsatzes:

- Programme in CFC sind weniger fehleranfällig. Keine doppelten Übergabemerker!

- Die Fehlersuche in CFC Programmen ist leichter, weil mehrere Bausteine online mit allen Parametern beobachtbar sind. Sie brauchen keine „Steuern Variablen“ Funktion.

- Programme sind transparenter und leichter erweiterbar. Sie sehen den Zusammenhang der Aufgabenstellung deutlicher im CFC Planeditor als in AWL/FUP/KOP Programmen.

- Das Original der CFC Pläne ist von enormer Bedeutung. Ohne dieses Original ist eine Rückinterpretation nicht möglich.

- Verfahrenstechnische oder strukturierte Aufgabenstellungen lassen sich wesentlich effektiver mit CFC Plan lösen. Der Endkunde erhält eine Lösung, die er selber versteht!

- Programme in CFC weisen einen höheren Speicherbedarf auf. Dies ist kein wesentliches negatives Argument.

- Die Zykluszeit Belastung ist bei CFC Programme höher. (Es ist der Einsatz einer größeren CPU zu prüfen)

- Es wird eine zusätzliche Lizenzen (CFC) benötigt

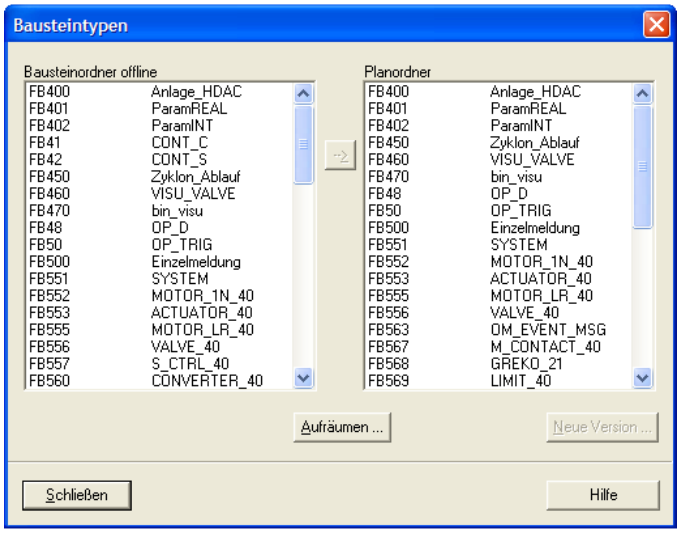
- Die Online Änderbarkeit von Programmen ist etwas aufwendiger, da die Programme immer erst compiliert werden müssen, bevor diese in das AG geladen werden können.

- Die automatische Variablengenerierung (Mappen) ist ein erheblicher Vorteil der CFC Plantechnik. Nicht jeder Text muss in einen DB gelegt werden!

- Beim Compilieren der CFC Pläne werden die Instanz DB`s automatisch neu zugeordnet. Programme mit Operator Panels müssen dies berücksichtigen. Es ist ein Übergabe DB notwendig.

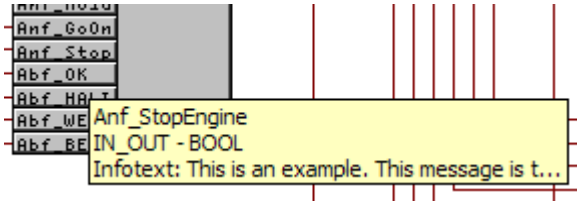
Vom Anwender erstellte Programme können mühelos in CFC Pläne integriert werden.

Bausteine im Bausteinordner ➔ Bausteine im Planordner



Hinweise:

Parameternamen von Bausteinen sollten in den ersten 8 Zeichen eindeutig sein! Nur diese ersten Zeichen werden im CFC Plan sichtbar. Alle weiteren Zeichen sind nur als Tooltip Text sichtbar.



Vom Eingang „Anf_StopEngine“ ist nur der Text „Anf_Stop“ lesbar. Der Rest ist nur über den Tooltip Text erkennbar.

Der Kommentar des Eingangs ist zu lang, da im CFC Plan nicht alle Zeichen (50 Zeichen) ausgegeben werden. Der Programmierer hat darauf zu achten, dass alle zur Identifikation notwendigen Zeichen lesbar sind.

18.2 Programmerstellung mit AWL/FUP/KOP

Warum AWL, FUP, KOP

Der Grund für die Erstellung eines Programms in FUP und KOP basierte auf der Betrachtung, dass Elektriker die Programme lesen können. Die Erweiterung, das AWL als Erstellersprache zugelassen war, war in der Vergangenheit eher die Ausnahme. Die Komplexität der Programmlösungen ist jedoch derart stark gestiegen, dass viele Programme nicht mehr durch die Zielgruppe Elektriker (Betriebsbetreuer) lesbar sind. Es sind heute Spezialisten gefordert, die die Programme erstellen und pflegen. Dies führt immer wieder zu der Problematik, dass eine große Abhängigkeit zwischen Lieferant und Auftraggeber besteht und damit keine Kostensicherheit existiert.

Vorteile CFC gegenüber AWL, FUP, KOP Programmen

Programme in AWL/FUP/KOP sind fehleranfälliger als CFC-Programme, weil Verbindungsmerker doppelt vergeben werden können.

Programmierer erstellen nicht nachvollziehbare Programme. Der Auftrageber hat eine starke Abhängigkeit zum Ersteller der Software.

Die Onlinebeobachtung von mehreren Netzwerken ist möglich

Vorteile AWL, FUP, KOP gegenüber CFC Programmen

AWL/FUP/KOP Programme sind gut geeignet bei Aufgabenstellungen mit geringer Komplexität und bei Aufgabenstellungen, die keine Bausteinbibliotheken beinhalten, die komplexe Funktionen realisieren (Regler, Dosierer, Umrichter,…)

Programme, die in AWL, FUP oder KOP erstellt wurden, weisen einen geringeren Speicherbedarf als CFC Programme auf.

Die Zykluszeitbelastung ist bei AWL/FUP/KOP Programme geringfügig niedriger als die der CFC-Programme.

AWL/FUP/KOP Programme sind ohne zusätzliche Lizenzen (CFC) erstellbar

Vorteile FUP gegenüber AWL Programmen

Die Online Beobachtung von FUP Programmen ist übersichtlicherer als die von AWL Programmen, da der Signalzustand eindeutig angezeigt wird.

Vorteile AWL gegenüber FUP Programmen

Die Symbolik in AWL Programmen ist immer in einer Zeile lesbar. AWL Programme sind komprimierter und komplexe Zusammenhänge lassen sich in AWL übersichtlicher lösen als in FUP-Programmen.

18.3 Bedienberechtigungsstufen für Faceplateanwahl (z.B. UDO)

Bedienung	Eigenschaftsname Deutsch (frei)	Eigenschaftsname Englisch (frei)	Attribut (nicht ändern!)
		VALVE	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson5
MSR	Berechtigung MSR	Permission PCE	pPermisson6
		A_INPUT	
OG2	Berechtigung OG2	Permission HL2	pPermisson1
OG1	Berechtigung OG1	Permission HL1	pPermisson2
UG1	Berechtigung UG1	Permission LL1	pPermisson3
UG2	Berechtigung UG2	Permission LL2	pPermisson4
XABS	Berechtigung XABS	Permission PV	pPermisson5
XINB	Berechtigung XINB	Permission PV_SIMU	pPermisson6
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson7
		MOTOR_1N:	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson5
MSR	Berechtigung MSR	Permission PCE	pPermisson6
		A_OUTPUT	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson5
MSR	Berechtigung MSR	Permission PCE	pPermisson6
Y	Berechtigung Y	Permission MV	pPermisson7
		C_CTRL	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson5
Y	Berechtigung Y	Permission MV	pPermisson7
WIWE	Berechtigung WIWE	Permission SP_EXT	pPermisson8
YHYA	Berechtigung YHYA	Permission Y_MV_MAN_AUT	pPermisson9
W	Berechtigung W	Permission SP	pPermisson10
OG2	Berechtigung OG2	Permission HL2	pPermisson11
OG1	Berechtigung OG1	Permission HL1	pPermisson12
UG1	Berechtigung UG1	Permission LL1	pPermisson13
UG2	Berechtigung UG2	Permission LL2	pPermisson14
XABS	Berechtigung XABS	Permission PV	pPermisson15
XINB	Berechtigung XINB	Permission PV_SIM	pPermisson16
		CONVERTER	
RL	Berechtigung RL	Permission RL	pPermisson1
LL	Berechtigung LL	Permission LL	pPermisson2
AUS	Berechtigung AUS	Permission OFF	pPermisson3
HAND	Berechtigung HAND	Permission MAN	pPermisson4
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson5
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson6
MSR	Berechtigung MSR	Permission PCE	pPermisson7
Y	Berechtigung Y	Permission MV	pPermisson8
		DIFF_INT:	
RESET	Berechtigung RESET	Permission RESET	pPermisson1
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson2
		FW_CONTROL:	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
HALT	Berechtigung HALT	Permission HOLD	pPermisson5

WEITER	Berechtigung WEITER	Permission CONTINUE	pPermisson6
FW	Berechtigung FW	Permission FW	pPermisson7
SW1	Berechtigung SW1	Permission SP1	pPermisson11
SW2	Berechtigung SW2	Permission SP2	pPermisson12
SW3	Berechtigung SW3	Permission SP3	pPermisson13
SW4	Berechtigung SW4	Permission SP4	pPermisson14
SW5	Berechtigung SW5	Permission SP5	pPermisson15
BW1	Berechtigung BW1	Permission BV1	pPermisson16
BW2	Berechtigung BW2	Permission BV2	pPermisson17
BW3	Berechtigung BW3	Permission BV3	pPermisson18
BW4	Berechtigung BW4	Permission BV4	pPermisson19
BW5	Berechtigung BW5	Permission BV5	pPermisson20
		DOS	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
W	Berechtigung W	Permission SP	pPermisson5
ZR	Berechtigung ZR	Permission RESET	pPermisson6
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson7
		SPLIT_RANGE:	
		AEM_VALVE	
Y	Berechtigung Y	Permission MV	pPermisson1
YMIN_ZULUF	Berechtigung YMIN_ZULUFT	Permission MV_MIN	pPermisson2
		SCHEDULE:	
Schedule	Berechtigung Schedule	Permission Schedule	pPermisson1
Exception	Berechtigung Exception	Permission Exception	pPermisson1
		RUN_TIME_METER:	
RTM_VALUE	Berechtigung RTM_VALUE	Permission RTM_VALUE	pPermisson1
		MOTOR_2N:	
N1	Berechtigung N1	Permission N1	pPermisson1
N2	Berechtigung N2	Permission N2	pPermisson2
AUS	Berechtigung AUS	Permission OFF	pPermisson3
HAND	Berechtigung HAND	Permission MAN	pPermisson4
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson5
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson6
		MOTOR_LR:	
RECHTS	Berechtigung RECHTS	Permission RIGHT	pPermisson1
LINKS	Berechtigung LINKS	Permission LEFT	pPermisson2
AUS	Berechtigung AUS	Permission OFF	pPermisson3
HAND	Berechtigung HAND	Permission MAN	pPermisson4
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson5
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson6
		S_CTRL	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson5
Y	Berechtigung Y	Permission MV	pPermisson7
WIWE	Berechtigung WIWE	Permission SP_INT/EXT	pPermisson8
YHYA	Berechtigung YHYA	Permission MV_MAN/AUT	pPermisson9
W	Berechtigung W	Permission SP	pPermisson10
OG2	Berechtigung OG2	Permission HL2	pPermisson11
OG1	Berechtigung OG1	Permission HL1	pPermisson12
UG1	Berechtigung UG1	Permission LL1	pPermisson13
UG2	Berechtigung UG2	Permission LL2	pPermisson14
XABS	Berechtigung XABS	Permission PV	pPermisson15
XINB	Berechtigung XINB	Permission PVSIM	pPermisson16
		ACTUATOR:	
EIN	Berechtigung EIN	Permission ON	pPermisson1
AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
HALT	Berechtigung HALT	Permission HOLD	pPermisson5
Y	Berechtigung Y	Permission MV	pPermisson6
SIMU	Berechtigung SIMU	Permission SIMU	pPermisson7
		CLOCK_TIMER:	
EIN	Berechtigung EIN	Permission ON	pPermisson1

AUS	Berechtigung AUS	Permission OFF	pPermisson2
HAND	Berechtigung HAND	Permission MAN	pPermisson3
AUTO	Berechtigung AUTO	Permission AUTO	pPermisson4
PARA	Berechtigung PARA	Permission PARA	pPermisson5

18.4 Belegung der Zustandsworte Q_STATE

Der Ausgang Q_STATE der Sm@rtLib Bauteine gibt den Zustand der Bausteine in der Steuerung an. Die einzelnen Zustände sind als Bits in diesem Doppelwort abgelegt. Diese können z.B. zur Visualisierung in WinCC Bildern verwendet werden.

A_INPUT

zHL2_SUPR	BIT 0
zHL1_SUPR	BIT 1
zLL1_SUPR	BIT 2
zLL2_SUPR	BIT 3
zHL2_SUPR_MSG	BIT 4
zHL1_SUPR_MSG	BIT 5
zLL1_SUPR_MSG	BIT 6
zLL2_SUPR_MSG	BIT 7
zHL2_OP_EN	BIT 8
zHL1_OP_EN	BIT 9
zLL1_OP_EN	BIT 10
zLL2_OP_EN	BIT 11
zQ_HL2	BIT 12
zQ_HL1	BIT 13
zQ_LL1	BIT 14
zQ_LL2	BIT 15
zPCE	BIT 16
zSIMU	BIT 17
zQ_AI_FLT	BIT 18
zQ_XF_FLT	BIT 19

C_CTRL

(Q_STATE)

zAUT_Z	BIT 0
zMAN_Z	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT_Z	BIT 3
zQ_ON_Z	BIT 4
zQ_ACT_Z	BIT 5
zQ_SP_EXT_Z	BIT 6
zQ_MV_AUT_Z	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zMV_TRKM_Z	BIT 20
zSP_TRKM_Z	BIT 21
zAO_FLT_Z	BIT 26
zHOME_POS	BIT 27

(Q_AI_STATE)

zHL2_SUPR	BIT 0
zHL1_SUPR	BIT 1
zLL1_SUPR	BIT 2
zLL2_SUPR	BIT 3
zHL2_SUPR_MSG	BIT 4
zHL1_SUPR_MSG	BIT 5
zLL1_SUPR_MSG	BIT 6
zLL2_SUPR_MSG	BIT 7
zLH2_OP_EN	BIT 8
zLH1_OP_EN	BIT 9
zLL1_OP_EN	BIT 10
zLL2_OP_EN	BIT 11
zQ_HL2_Z	BIT 12
zQ_HL1_Z	BIT 13
zQ_LL1_Z	BIT 14
zQ_LL2_Z	BIT 15
zXINB_Z	BIT 16
zAI_FLT_Z	BIT 18
zXF_FLT_Z	BIT 19

ACTUATOR

zQ_STOP	BIT 0
zQ_ILOC	BIT 1
zQ_TOEB	BIT 2
zQ_ENAB	BIT 3
zSWITCHLOCK	BIT 4
zBI_TIME	BIT 5
zHOLD	BIT 6
zFLT_CLS	BIT 7
zAUT	BIT 8
zMAN	BIT 9
zAUTO_ON	BIT 10
zQ_MAN_AUT	BIT 11
zQ_ON	BIT 12
zGOTO_OFF	BIT 13
zQ_ILOC_FLT	BIT 14
zQ_FB_FLT	BIT 15
zBI_OFF_T	BIT 16
zTORQ_SWITCHING	BIT 17
zAI_FLT	BIT 18
zQ_BI_FLT	BIT 19
zQ_FLT_EXT	BIT 20
zFB_CLS	BIT 21
zFB_OPN	BIT 22
zFB_OPN_MAX	BIT 23
zFB_CLS_MAX	BIT 24
zXF_FLT	BIT 25
zEMRG_STOP	BIT 26
zTRK	BIT 27
zQ_OPN_ACT	BIT 28
zQ_CLS_ACT	BIT 29
zQ_HALT_ACT	BIT 30

A_OUTPUT

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zQ_ACT	BIT 5
zEMRG_STOP	BIT 8
zSIMU	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zFB_CLS	BIT 21
zQ_FB_FLT	BIT 25
zQ_AO_FLT	BIT 26
zHOME_POS	BIT 27

DIFF_INT

zENABLE	BIT 0
zRESET_OP_EN	BIT 1
zTYP	BIT 2
zSIMU_Z	BIT 3

RTM

zENABLE	BIT 0

CONVERTER

(Q_STATE1)

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON1	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zAUTO_ON2	BIT 5
zQ_ACT1	BIT 6
zQ_ACT2	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zQ_BI_FLT	BIT 20
zQ_MS_FLT	BIT 21
zQ_CF_FLT	BIT 22
zQ_RDY_FLT	BIT 23
zSWITCHLOCK	BIT 24
zQ_FB_FLT	BIT 25
zQ_AO_FLT	BIT 26
zON1	BIT 27
zON2	BIT 28
zFB_ON1	BIT 29
zFB_ON2	BIT 30

(Q_STATE2)

zTRKM	BIT 0
zTRKV_ON1_2	BIT 1
zBI_OFF_T	BIT 2
zCONVERTER_ILOC	BIT 3

CLOCKTIMER

zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4

DOSER

zAUT_Z	BIT 0
zMAN_Z	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT_Z	BIT 3
zAUEI_Z	BIT 4
zANST_Z	BIT 5
zQ_EVRE_Z	BIT 6
zFrei	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zQ_FDS	BIT 20
zQ_GDS	BIT 21
zDOS_HOLD	BIT 22
zQ_RUN_IN_FLT_Z	BIT 26
zQ_DRIB_FLT_Z	BIT 27

MOTOR_2N

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON2	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zAUTO_ON1	BIT 5
zQ_ACT2	BIT 6
zQ_ACT1	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zQ_BI_FLT	BIT 20
zQ_MS_FLT	BIT 21
zTRKV_ON2	BIT 23
zTRKV_ON1	BIT 24
zQ_FB_FLT	BIT 25
zTRKM	BIT 26
zSWITCH_LOCK	BIT 27
zQ_BI_DT_ACT	BIT 28
zFB_MCC2_INTERNAL	BIT 29
zFB_MCC1_INTERNAL	BIT 30

FW_CONTROL

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zQ_ACT	BIT 5
zQ_HOLD	BIT 6
zQ_FLT	BIT 7
zEMRG_STOP	BIT 8
zSIMU	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zCOND_5	BIT 20
zFB_RUN	BIT 21
zCOND_6	BIT 22
zCOND_7	BIT 23
zCOND_8	BIT 24
zQ_FB_FLT	BIT 25
zQ_BV1	BIT 26
zQ_BV2	BIT 27
zQ_BV3	BIT 28
zQ_BV4	BIT 29
zQ_BV5	BIT 30

S_CTRL

(Q_STATE)	
zAUT_Z	BIT 0
zMAN_Z	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT_Z	BIT 3
zQ_ON_Z	BIT 4
zQ_ACT_Z	BIT 5
zQ_SP_EXT_Z	BIT 6
zQ_MV_AUT_Z	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zMV_TRKM_Z	BIT 20
zSP_TRKM_Z	BIT 21
zQ_HIGH_ACT	BIT 22
zQ_LOW_ACT	BIT 23
zFB_HIGH	BIT 24
zFB_LOW	BIT 25
(Q_AI_STATE)	
zHL2_SUPR	BIT 0
zHL1_SUPR	BIT 1
zLL1_SUPR	BIT 2
zLL2_SUPR	BIT 3
zHL2_SUPR_MSG	BIT 4
zHL1_SUPR_MSG	BIT 5
zLL1_SUPR_MSG	BIT 6
zLL2_SUPR_MSG	BIT 7
zLH2_OP_EN	BIT 8
zLH1_OP_EN	BIT 9
zLL1_OP_EN	BIT 10
zLL2_OP_EN	BIT 11
zQ_HL2_Z	BIT 12
zQ_HL1_Z	BIT 13
zQ_LL1_Z	BIT 14
zQ_LL2_Z	BIT 15
zXINB_Z	BIT 16
zAI_FLT_Z	BIT 18
zXF_FLT_Z	BIT 19

MOTOR_1N

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zQ_ACT	BIT 5
zTRK	BIT 6
zTRK_ON	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zFB_MCC	BIT 20
zQ_MS_FLT	BIT 21
zQ_BI_FLT	BIT 22
zQ_BI_OFF_END	BIT 23
zFB_MCC_EN	BIT 24
zQ_MCC_FLT	BIT 25

VALVE

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ON	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zQ_ACT	BIT 5
zTRK	BIT 6
zTRK_ON	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zFB_ACT_ON	BIT 20
zFB_ACT_OFF	BIT 21
zQ_FB_OPN	BIT 22
zQ_FB_CLS	BIT 23
zHOME_POS	BIT 24
zQ_FB_FLT	BIT 25

SCHEDULE

zWEEK_PROG_EN	BIT 0
zEXCPT_PROG_EN	BIT 1
zQ_OFF	BIT 2
zQ_ON_LEVEL1	BIT 3
zQ_ON_LEVEL2	BIT 4
zAusnahmeAKTIV	BIT 5

MOTOR_LR

zAUT	BIT 0
zMAN	BIT 1
zAUTO_ONL	BIT 2
zQ_MAN_AUT	BIT 3
zQ_ON	BIT 4
zAUTO_ONR	BIT 5
zQ_ACTL	BIT 6
zQ_ACTR	BIT 7
zEMRG_STOP	BIT 8
zSIMU_Z	BIT 9
zCOND_SUPR	BIT 10
zQ_ILOC_FLT	BIT 11
zCOND_1	BIT 12
zCOND_2	BIT 13
zCOND_3	BIT 14
zCOND_4	BIT 15
zQ_STOP	BIT 16
zQ_ILOC	BIT 17
zQ_TOEB	BIT 18
zQ_ENAB	BIT 19
zQ_BI_FLT	BIT 20
zQ_MS_FLT	BIT 21
zTRKV_ONL	BIT 23
zTRKV_ONR	BIT 24
zQ_FB_FLT	BIT 25
zTRKM	BIT 26
zSWITCH_LOCK	BIT 27
zQ_BI_DT_ACT	BIT 28
zFB_MCCL_INTERNAL	BIT 29
zFB_MCCR_INTERNAL	BIT 30

18.5 Bediennummern

Funktionsbeschreibung	Bediennummer CMD	Begleitwert CMD_V	intern
Umschalten Betriebsart Hand	1	-	HAND
Umschalten Betriebsart Automatik	2	-	AUTO
Element Ausschalten	3	-	AUS
Element Einschalten	4	-	EIN
Element Quittieren	5	-	QS
Umschaltung Sollwert intern/extern (toggeln)	6	-	WIWE
Umschaltung Stellgrösse Hand/Automatik (toggeln)	7	-	YHYA
Eingabe Stellgrösse	8	x	Y
Eingabe Sollwert	10	x	W
Eingabe Grenzwert Oben 2	14	x	OG2
Eingabe Grenzwert Oben 1	15	x	OG1
Eingabe Grenzwert Unten 1	16	x	UG1
Eingabe Grenzwert Unten 2	17	x	UG2
Eingabe Hysterese	0	x	HYSTERESE
Eingabe Istwert	19	x	ISTWERT
Umschaltung Inbetriebnahmeistwert ein/aus (toggeln)	18	-	XINB
Eingabe Kp-Wert	20	x	KP
Eingabe Tn-Wert	21	x	TN
Eingabe Tv-Wert	22	x	TV
Element Einschalten Linkslauf	23	-	LL
Element Einschalten Rechtslauf	24	-	RL
Rücksetzen	25	-	RESET
Element anhalten	26	-	HALT
Element Fortsetzen	27	-	WEITER
Eingabe Arbeitspunkt	28	x	AP
Eingabe Sollwert 1	30	x	SW1
Eingabe Sollwert 2	31	x	SW2
Eingabe Sollwert 3	32	x	SW3
Eingabe Sollwert 4	33	x	SW4
Eingabe Sollwert 5	34	x	SW5
Eingabe Sollwert 6	35	x	SW6
Eingabe Sollwert 7	36	x	SW7
Eingabe Sollwert 8	37	x	SW8
Umschalten Binärwert 1	40	-	BW1
Umschalten Binärwert 2	41	-	BW2
Umschalten Binärwert 3	42	-	BW3
Umschalten Binärwert 4	43	-	BW4
Umschalten Binärwert 5	44	-	BW5
Umschalten Binärwert 6	45	-	BW6
Umschalten Binärwert 7	46	-	BW7
Umschalten Binärwert 8	47	-	BW8
Anwahl Fahrweise 1	51	-	FW1
Anwahl Fahrweise 2	52	-	FW2
Anwahl Fahrweise 3	53	-	FW3
Anwahl Fahrweise 4	54	-	FW4
Anwahl Fahrweise 5	55	-	FW5
Hinweis im Faceplate wurde quittiert (Hinweis schließen)	99	-	MELDUNG
Element in Simulationsmodus ein,- ausschalten (toggeln)	50	-	SIMULATION
Element Einschalten Drehzahl 1	81	-	N1
Element Einschalten Drehzahl 2	82	-	N2
Element Einschalten Rechts	83	-	RECHTS
Element Einschalten Links	84	-	LINKS
Eingabe Totband	100	x	DEADB
Eingabe Pulszeit	101	x	PULSE_TM
Eingabe Pausenzeit	102	x	BREAK_TM
Eingabe Motorstellzeit	103	x	MTR_TM
Plus Impuls ausgeben	104	-	PLUS
Minus Impuls ausgeben	105	-	MINUS
Eingabe Impulslänge für Stellen Tippbetrieb	106	X	MAN_PULS_TM
Eingabe Stellgrössenrampe für Tippbetrieb	107	x	MV_RAMP
Element Suchfahrt	110	-	SUCHFAHRT
Artikelnummer Vorförderer übernehmen	111	-	ARTIKELNRCPY
Position 1	121	-	POS1

Position 2	122	-	POS2
Position 3	123	-	POS3
Position 4	124	-	POS4
Eingabe Mindestpulszeit	106	x	MV_PULS_TM
Eingabe Reihenfolge (Pumpenmanager)	150	X	
Reihenfolge Umschalten	151	-	
Eingabe Mindestmenge Zuluft	301	x	YMIN_ZULUFT
Scheduler: Startzeit Sonntag	1001	x	SUF
Scheduler: Stopzeit Sonntag	1002	x	SUT
Scheduler: Stufe Sonntag	1003	x	SUPL
Scheduler: Startzeit Montag	1004	x	MOF
Scheduler: Stopzeit Montag	1005	x	MOT
Scheduler: Stufe Montag	1006	x	MOPL
Scheduler: Startzeit Dienstag	1007	x	TUF
Scheduler: Stopzeit Dienstag	1008	x	TUT
Scheduler: Stufe Dienstag	1009	x	TUPL
Scheduler: Startzeit Mittwoch	1010	x	WEF
Scheduler: Stopzeit Mittwoch	1011	x	WET
Scheduler: Stufe Mittwoch	1012	x	WEPL
Scheduler: Startzeit Donnerstag	1013	x	THF
Scheduler: Stopzeit Donnerstag	1014	x	THT
Scheduler: Stufe Donnerstag	1015	x	THPL
Scheduler: Startzeit Freitag	1016	x	FRF
Scheduler: Stopzeit Freitag	1017	x	FRT
Scheduler: Stufe Freitag	1018	x	FRPL
Scheduler: Startzeit Samstag	1019	x	SAF
Scheduler: Stopzeit Samstag	1020	x	SAT
Scheduler: Stufe Samstag	1021	x	SAPL
Scheduler: Ausnahme Start	1031	x	EX_FROM
Scheduler: Ausnahme Ende	1032	x	EX_TO
Scheduler: Ausnahme aktivieren	1033	x	EX_ON
Scheduler: Ausnahme deaktivieren	1034	x	EX_OFF
Scheduler: Wochenprogramm aktivieren	1035	x	DAY_SCHEDULE_ON
Scheduler: Wochenprogramm deaktivieren	1036	x	DAY_SCHEDULE_OFF
Eingabe Betriebsstundenzähler	1050	x	RTM_VALUE
ClockTimer: Segment aktivieren Sonntag	1061	x	SUON
ClockTimer: Segment deaktivieren Sonntag	1062	x	SUOFF
ClockTimer: Segment aktivieren Montag	1063	x	MOON
ClockTimer: Segment deaktivieren Montag	1064	x	MOOFF
ClockTimer: Segment aktivieren Dienstag	1065	x	TUON
ClockTimer: Segment deaktivieren Dienstag	1066	x	TUOFF
ClockTimer: Segment aktivieren Mittwoch	1067	x	WEON
ClockTimer: Segment deaktivieren Mittwoch	1068	x	WEOFF
ClockTimer: Segment aktivieren Donnerstag	1069	x	THON
ClockTimer: Segment deaktivieren Donnerstag	1070	x	THOFF
ClockTimer: Segment aktivieren Freitag	1071	x	FRON
ClockTimer: Segment deaktivieren Freitag	1072	x	FROFF
ClockTimer: Segment aktivieren Samstag	1073	x	SAON
ClockTimer: Segment deaktivieren Samstag	1074	x	SAOFF

18.6 Festlegungen zur Visualisierung

Monitor	19“ TFT Monitor mit 800mm Sehabstand
Bildschirmauflösung	Allgemein 1280 * 1024 [Bildgröße 60,5 x 72,6 bei DeltaV]
Hintergrundfarbe	RGB : 192, 192, 192
Objekte im Bild	RGB : 128, 128, 128 mit einem Rahmen von 2 Pixel in schwarz
Überschrift im Bild	RGB : 128, 128, 128 Arial 28
Kennzeichnung von PLT Stellen	RGB : 128, 128, 128 Arial 10
Kennzeichnung von Behältern	RGB : 128, 128, 128 Arial 14
Wertanzeigen im Bild (Nachkommastellen)	Die Anzeige soll 0,1 % vom Messbereich anzeigen
Hauptprodukt oder Versorgungsleitungen	Siehe Farbfestlegung der Leitungen 5 Pixel Volllinie
Nebenprodukt oder Nebenversorgungsleitungen	Siehe Farbfestlegung der Leitungen 3 Pixel Volllinie
Anschlussleitung Versorgung (Sole, Dampf, Wasser, Energien)	Siehe Farbfestlegung der Leitungen 3 Pixel Volllinie
Wirklinien	RGB : 255,255,255 (Schwarz) 1 Pixel gestrichelt mit grauem Hintergrund Es müssen Wirklinien für Regelkreise integriert werden. Diese müssen in einen separaten Layer projektiert werden. Dies erleichtert ein eventuelles Ausblenden dieser Linien. Wirklinien müssen eine Wirkrichtung in Form eines Pfeils enthalten
Schriftgröße Informationstexte oder Beschriftung	Mindestens Arial 10
Schriftgröße Wertanzeigen ohne dauernde Beobachtung	Mindestens Arial 14
Schriftgröße Wertanzeigen mit dauernder Beobachtung	Mindestens Arial 18

Beispiel für die Auswahl von Leitungsfarben

Stoffgruppe	Farbe	RAL	RGB
Wasser	Grün	6018	079, 168, 051
Wasserdampf	Rot	3000	171, 031, 028
Luft	Grau	7001	135, 148, 166
Säure	Orange	2003	255, 099, 054
Laugen	Violett	4001	130, 064, 128
Sauerstoff	Blau	5015	023, 097, 171
Flüssigkeiten (alle anderen)	Braun	8001	145, 082, 046
Gase (alle anderen)	Gelb	1021	252, 189, 031